

Analysis and Computation of the Joint Queue Length Distribution in a FIFO Single-Server Queue with Multiple Batch Markovian Arrival Streams*

Hiroyuki Masuyama[†] and Tetsuya Takine[‡]

Abstract

This paper considers a work-conserving FIFO single-server queue with multiple batch Markovian arrival streams governed by a continuous-time finite-state Markov chain. A particular feature of this queue is that service time distributions of customers may be different for different arrival streams. After briefly discussing the actual waiting time distributions of customers from respective arrival streams, we derive a formula for the vector generating function of the time-average joint queue length distribution in terms of the virtual waiting time distribution. Further assuming the discrete phase-type batch size distributions, we develop a numerically feasible procedure to compute the joint queue length distribution. Some numerical examples are provided also.

Keywords: Single-server queue; FIFO; Batch Markovian arrival streams; Joint queue length.

Mathematics Subject Classification: Primary 60K25; Secondary 60J22

I Introduction

In this paper, we study the joint queue length distribution in a stationary work-conserving FIFO single-server queue fed by multiple batch arrival streams governed by a continuous-time finite-state Markov chain. A particular feature of this queue is that service time distributions of customers may be different for different arrival streams.

Single-server queues with Markovian arrival streams have been extensively studied for last two decades. At present, the most popular Markovian arrival stream is MAP (Markovian arrival process) introduced in [6]. MAP is a class of semi-Markovian arrival processes including Markov modulated Poisson processes and phase-type renewal processes as special cases. After

*This paper is published in Stochastic Models, vol. 19, no. 3, pp. 349-381, 2003.

[†]Dept. of Systems Science, Graduate School of Informatics, Kyoto University;

E-mail: masuyama@sys.i.kyoto-u.ac.jp

[‡]Dept. of Information and Communications Technology, Graduate School of Engineering, Osaka University;

E-mail: takine@comm.eng.osaka-u.ac.jp

introducing MAP, some extensions have been made. One is batch MAP [7] that allows batch arrivals and the other is marked MAP [2, 3, 4] that explicitly represents possibly correlated multiple Markovian arrival streams. The arrival process in this paper has these two features, i.e., batch marked MAP.

Most of previous works on FIFO single-server queues with Markovian arrival streams assume that service times of all customers are independent and identically distributed (i.i.d.) according to a common distribution function. As a result, the bivariate process of the total number of customers and the state of the Markov chain that governs the arrival process immediately after departures forms a Markov chain of M/G/1 type and the steady-state solution can be computed by well-known M/G/1 paradigm [9].

On the other hand, if service time distributions of customers from respective arrival streams are different from one another, the bivariate process does not have the Markov property [17], except for queues with a superposition of independent Poisson streams. Thus the queue length analysis of such a queue is not straightforward. Note, however, that the virtual waiting time process in such a queue is characterized by a bivariate Markov process [1, 11, 13, 18], and algorithmic solution methods are known in the literature [11, 13].

Recently, a new approach was developed to characterize the joint queue length distribution in FIFO queues with marked MAP having different service time distributions [16, 17]. In these works, the invariant relationship of the joint queue length distributions at a random point in time and at departures was obtained and from this, the distributional form of Little's law was established in [16]. Further, based on the latter, an algorithmic solution method was developed [16, 17]. Related works are found in [8, 10]. See [15] for a survey of those developments.

The results in this paper are considered as an extension of those in [17], allowing batch arrivals in each arrival stream. Note here that the distributional form of Little's law does not hold for FIFO queues with batch arrivals. Therefore our starting point in analyzing the time-average joint queue length distribution is the invariant relationship of the joint queue length distributions at a random point in time and at departures in [16]. By doing so, the problem is reduced to find the joint queue length distributions at departures of customers from respective arrival streams.

As you will see, the joint queue length distribution at departures in the FIFO queue is closely related to the virtual waiting time distribution that is readily obtained with the known results. Using these facts, we derive a general formula for the stationary joint queue length distribution at departures in terms of the sojourn time distribution. Further, assuming discrete phase-type batch size distributions, we derive recursions to compute the joint queue length distribution.

The above outline is similar to the single arrival case in [17]. However, the implementation of some of those recursions is not trivial, because we have to determine several truncation and stopping criteria, which are due to batch arrivals, and their straightforward implementation

would require very huge memory space and time-consuming. In this paper, assuming discrete phase-type batch size distributions, we propose a numerically feasible procedure to compute those recursions, while ensuring the numerical accuracy in the final result. This is the main contribution of this paper. Note that our procedure is applicable to the FIFO BMAP/G/1 queue with i.i.d. services. too, when the batch size distribution follows a discrete phase-type distribution.

The rest of this paper is divided into six sections. In section II, the mathematical model is described. In section III, we briefly discuss the virtual and actual waiting time distributions. In section IV, we first derive a general formula for the joint queue length distribution, and assuming the discrete phase-type batch sizes, we show recursive formulas to compute the joint queue length distribution. In section V, the implementation of the recursions is discussed. In section VI, we discuss the efficiency of our algorithm and the qualitative behavior of the queue length through some numerical examples. Finally, concluding remarks are provided in section VII. Throughout the paper, we denote matrices and vectors by bold capital letters and bold small letters, respectively.

II Model

We consider a work-conserving FIFO single-server queue fed by K arrival streams. We call customers arriving from the k th ($k = 1, \dots, K$) arrival stream class k customers. Let \mathcal{K} denote a set of class indices, i.e., $\mathcal{K} = \{1, 2, \dots, K\}$.

Customer arrivals are governed by a continuous-time Markov chain, which is called the underlying Markov chain hereafter. The underlying Markov chain has a finite state space $\mathcal{M} = \{1, \dots, M\}$ and it is assumed to be irreducible. The underlying Markov chain stays in state $i \in \mathcal{M}$ for an exponential interval of time with mean μ_i^{-1} . When the sojourn time in state i has elapsed, with probability $\sigma_{i,j}(0)$ ($j \in \mathcal{M}, j \neq i$), the underlying Markov chain changes its state to state j without arrivals. Also, with probability $\sigma_{k,i,j}(n)$ ($k \in \mathcal{K}, n = 1, 2, \dots$), the underlying Markov chain changes its state to state j and n customers of class k arrive simultaneously. For convenience, let $\sigma_{i,i}(0) = 0$ for all $i \in \mathcal{M}$. Then

$$\sum_{j \in \mathcal{M}} \left(\sigma_{i,j}(0) + \sum_{k \in \mathcal{K}} \sum_{n=1}^{\infty} \sigma_{k,i,j}(n) \right) = 1,$$

for all $i \in \mathcal{M}$. We assume that service times of class k ($k \in \mathcal{K}$) customers are i.i.d. according to a distribution function $H_k(x)$ with finite mean h_k .

We now introduce some notations to describe the above arrival process. Let \mathbf{C} denote an

$M \times M$ matrix whose (i, j) th $(i, j \in \mathcal{M})$ element $C_{i,j}$ is given by

$$C_{i,j} = \begin{cases} -\mu_i, & \text{if } i = j, \\ \sigma_{i,j}(0)\mu_i, & \text{otherwise.} \end{cases}$$

Further, for $k \in \mathcal{K}$, we define $\mathbf{D}_k(n)$ ($n = 1, 2, \dots$) as an $M \times M$ matrix whose (i, j) th $(i, j \in \mathcal{M})$ element $D_{k,i,j}(n)$ is given by

$$D_{k,i,j}(n) = \sigma_{k,i,j}(n)\mu_i.$$

Thus the counting process of arrivals is characterized by the set of matrices $(\mathbf{C}, \mathbf{D}_1(n_1), \dots, \mathbf{D}_K(n_K))$. Roughly speaking, customers arrive in the following way. When a state transition driven by $\mathbf{D}_k(n)$ occurs, n customers of class k arrive simultaneously. On the other hand, when a state transition driven by \mathbf{C} occurs, no customers arrive.

We define \mathbf{D}_k ($k \in \mathcal{K}$) and \mathbf{D} as

$$\mathbf{D}_k = \sum_{n=1}^{\infty} \mathbf{D}_k(n), \quad \mathbf{D} = \sum_{k \in \mathcal{K}} \mathbf{D}_k,$$

respectively. Note that the infinitesimal generator of the underlying Markov chain is given by $\mathbf{C} + \mathbf{D}$. Note also that $(\mathbf{C} + \mathbf{D})\mathbf{e} = \mathbf{0}$, where \mathbf{e} denotes a column vector whose elements are all equal to one. We denote, by $\boldsymbol{\pi}$, the stationary probability vector of the underlying Markov chain and therefore $\boldsymbol{\pi}(\mathbf{C} + \mathbf{D}) = \mathbf{0}$ and $\boldsymbol{\pi}\mathbf{e} = 1$. Because of the finite state space \mathcal{M} and the irreducibility of the underlying Markov chain, $\boldsymbol{\pi}$ is uniquely determined.

We define λ_k ($k \in \mathcal{K}$) as

$$\lambda_k = \sum_{n=1}^{\infty} n \boldsymbol{\pi} \mathbf{D}_k(n) \mathbf{e}.$$

Note that λ_k denotes the arrival rate of class k customers, i.e., the mean number of class k customers arriving in a unit time in steady state. We assume that at least one element of \mathbf{D}_k ($k \in \mathcal{K}$) is positive, so that $\lambda_k > 0$ for all $k \in \mathcal{K}$. Let ρ_k denote the utilization factor of class k customers, i.e.,

$$\rho_k = \lambda_k h_k, \quad k \in \mathcal{K}.$$

Furthermore, we denote the overall arrival rate by $\lambda = \sum_{k \in \mathcal{K}} \lambda_k$ and the overall utilization factor by $\rho = \sum_{k \in \mathcal{K}} \rho_k$. In the remainder of this paper, we assume that $\rho < 1$, which ensures that all customers arriving to the system are eventually served [5].

III Waiting Time Distribution

In this section, we consider the stationary distribution of the actual waiting time. To do so, we first consider the virtual waiting time that is equivalent to the amount of work in system. Let V

denote a generic random variable representing the stationary amount of work in system (i.e., the total amount of unfinished services of all customers in the system). Also let S denote a generic random variable representing the state of the underlying Markov chain in steady state. We then define $\mathbf{v}(x)$ as a $1 \times M$ vector whose j th element represents $\Pr[V \leq x, S = j]$. The Laplace-Stieltjes transforms (LSTs) of $H_k(x)$ and $\mathbf{v}(x)$ are denoted by $H_k^*(s)$ and $\mathbf{v}^*(s)$, respectively.

We define $\mathbf{D}(x)$ as

$$\mathbf{D}(x) = \sum_{k \in \mathcal{K}} \sum_{n=1}^{\infty} \mathbf{D}_k(n) H_k^{(n)}(x), \quad x \geq 0,$$

where $H_k^{(1)}(x) = H_k(x)$ and $H_k^{(n)}(x)$ ($n = 2, 3, \dots$) denotes the n -fold convolution of $H_k(x)$ with itself. Let \mathbf{Q} denote an $M \times M$ matrix that represents the infinitesimal generator of the underlying Markov chain obtained by excising the busy periods [11]. Note that \mathbf{Q} satisfies

$$\mathbf{Q} = \mathbf{C} + \int_0^{\infty} d\mathbf{D}(x) \exp(\mathbf{Q}x).$$

Let $\boldsymbol{\kappa}$ denote a $1 \times M$ vector that satisfies

$$\boldsymbol{\kappa} \mathbf{Q} = \mathbf{0}, \quad \boldsymbol{\kappa} \mathbf{e} = 1.$$

Applying the results in [11] to our model, we obtain the following theorem.

Theorem III.1 ([11]) $\mathbf{v}(0)$ is given by

$$\mathbf{v}(0) = (1 - \rho) \boldsymbol{\kappa}.$$

Furthermore, the LST $\mathbf{v}^*(s)$ of $\mathbf{v}(x)$ satisfies

$$\mathbf{v}^*(s) [s\mathbf{I} + \mathbf{C} + \mathbf{D}^*(s)] = s(1 - \rho) \boldsymbol{\kappa}, \quad \text{Re}(s) > 0, \quad (1)$$

where $\mathbf{D}^*(s)$ denotes the LST of $\mathbf{D}(x)$:

$$\mathbf{D}^*(s) = \int_0^{\infty} e^{-sx} d\mathbf{D}(x) = \sum_{k \in \mathcal{K}} \sum_{n=1}^{\infty} \mathbf{D}_k(n) \{H_k^*(s)\}^n. \quad (2)$$

We now consider the actual waiting time of class k customers in steady state. We define $W_k(n; m)$ as a generic random variable representing the actual waiting time of a randomly chosen class k customer who is a member of a batch of size n and the m th served customer among members of the same batch. Let $S^{(\mathbf{A}_k)}(n)$ denote a generic random variable representing the state of the underlying Markov chain immediately after class k batches of size n arrive. With those, we define $\mathbf{w}_k(x | n; m)$ as a $1 \times M$ vector whose j th element represents $\Pr[W_k(n; m) \leq x, S^{(\mathbf{A}_k)}(n) = j]$. Note that $\mathbf{w}_k(x | n; 1)$ ($n \geq 1$) is given by [11]

$$\mathbf{w}_k(x | n; 1) = \frac{\mathbf{v}(x) \mathbf{D}_k(n)}{\pi \mathbf{D}_k(n) \mathbf{e}}, \quad (3)$$

and for $n \geq 2$ and $m = 2, 3, \dots, n$,

$$\mathbf{w}_k(x | n; m) = \int_0^x \mathbf{w}_k(x - y | n; 1) dH_k^{(m-1)}(y), \quad x \geq 0. \quad (4)$$

Let W_k and $S^{(A_k)}$ denote generic random variables representing the actual waiting time of class k customers and the state of the underlying Markov chain immediately after arrivals of class k batches, respectively. We then define $\mathbf{w}_k(x)$ as a $1 \times M$ vector whose j th element represents $\Pr[W_k \leq x, S^{(A_k)} = j]$. Because a randomly chosen customer of class k is a member of a batch of size n with probability $n\pi \mathbf{D}_k(n)e/\lambda_k$, we have

$$\mathbf{w}_k(x) = \sum_{n=1}^{\infty} \frac{n\pi \mathbf{D}_k(n)e}{\lambda_k} \cdot \frac{1}{n} \sum_{m=1}^n \mathbf{w}_k(x | n; m). \quad (5)$$

Let $\mathbf{w}_k^*(s)$ denote the LST of $\mathbf{w}_k(x)$. From (3)–(5), we have

$$\mathbf{w}_k^*(s) = \sum_{n=1}^{\infty} \frac{\mathbf{v}^*(s) \mathbf{D}_k(n)}{\lambda_k} \sum_{m=1}^n \{H_k^*(s)\}^{m-1}, \quad \operatorname{Re}(s) > 0.$$

Thus we obtain the following theorem.

Theorem III.2 $\mathbf{w}_k^*(s)$ ($k \in \mathcal{K}$) is given by

$$\mathbf{w}_k^*(s) = \frac{\mathbf{v}^*(s) (\mathbf{D}_k - \mathbf{D}_k^*(H_k^*(s)))}{\lambda_k (1 - H_k^*(s))}, \quad \operatorname{Re}(s) > 0,$$

where

$$\mathbf{D}_k^*(z_k) = \sum_{n=1}^{\infty} z_k^n \mathbf{D}_k(n). \quad (6)$$

IV Joint Queue Length Distribution

This section considers the joint queue length distribution. In subsection IV.1, we apply a general relationship between the time-average queue length distribution and the queue length distributions at departures of customers of respective classes [16] to our specific queue. Then the problem is reduced to characterize the joint queue length distributions at departures of respective classes, which is discussed in subsection IV.2. Finally in subsection IV.3, assuming discrete phase-type batch size distributions, we derive recursions for some quantities required in computing the joint queue length distribution.

IV.1 Relationship in the joint queue length distributions

Let N_k ($k \in \mathcal{K}$) denote a generic random variable representing the number of class k customers in steady state. We define $\mathbf{p}(n_1, \dots, n_K)$ as a $1 \times M$ vector whose j th element represents

$\Pr[N_1 = n_1, \dots, N_K = n_K, S = j]$. For simplicity, let \mathbf{n} and \mathbf{z} denote a $1 \times K$ nonnegative integer vector (n_1, \dots, n_K) and a $1 \times K$ complex vector (z_1, \dots, z_K) , respectively. Further we define \mathcal{Z} as

$$\mathcal{Z} = \{(n_1, \dots, n_K); n_k = 0, 1, \dots, \text{ for all } k \in \mathcal{K}\}.$$

We then define $\mathbf{p}^*(\mathbf{z})$ as

$$\mathbf{p}^*(\mathbf{z}) = \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{p}(\mathbf{n}), \quad |z_k| \leq 1 \text{ for all } k \in \mathcal{K}.$$

Note that $\mathbf{p}^*(\mathbf{z})$ denotes the vector generating function of the joint queue length distribution in steady state.

Let $N_\nu^{(\mathcal{D}_k)}$ and $S^{(\mathcal{D}_k)}$ ($k, \nu \in \mathcal{K}$) denote generic random variables representing the number of class ν customers and the state of the underlying Markov chain, respectively, immediately after departures of class k customers in steady state. We then define $\mathbf{q}_k(\mathbf{n})$ ($k \in \mathcal{K}, \mathbf{n} \in \mathcal{Z}$) as a $1 \times M$ vector whose j th element represents $\Pr[N_1^{(\mathcal{D}_k)} = n_1, \dots, N_K^{(\mathcal{D}_k)} = n_K, S^{(\mathcal{D}_k)} = j]$. Further we define $\mathbf{q}_k^*(\mathbf{z})$ ($k \in \mathcal{K}$) as

$$\mathbf{q}_k^*(\mathbf{z}) = \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{q}_k(\mathbf{n}), \quad |z_k| \leq 1 \text{ for all } k \in \mathcal{K}.$$

Note that $\mathbf{q}_k^*(\mathbf{z})$ denotes the vector generating function of the joint queue length distribution immediately after departures of class k customers. Thus, applying Theorem 1 in [16] to our model, we have the following theorem.

Theorem IV.1 ([16]) $\mathbf{p}^*(\mathbf{z})$ and $\mathbf{q}_k^*(\mathbf{z})$ are related by

$$\mathbf{p}^*(\mathbf{z}) \left[\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z_k) \right] = \sum_{k \in \mathcal{K}} \lambda_k (z_k - 1) \mathbf{q}_k^*(\mathbf{z}), \quad (7)$$

where $\mathbf{D}_k^*(z_k)$ is given in (6).

Further, comparing the coefficient vectors of $z_1^{n_1} \cdots z_K^{n_K}$ on both sides of (7), we obtain the following result.

Corollary IV.1 The $\mathbf{p}(\mathbf{n})$ ($\mathbf{n} \in \mathcal{Z}$) is recursively determined by

$$\begin{aligned} \mathbf{p}(\mathbf{0}) &= \sum_{k \in \mathcal{K}} \lambda_k \mathbf{q}_k(\mathbf{0}) (-\mathbf{C})^{-1}, \\ \mathbf{p}(\mathbf{n}) &= \sum_{k \in \mathcal{K}} \left[\lambda_k (\mathbf{q}_k(\mathbf{n}) - \mathbf{q}_k(\mathbf{n} - \mathbf{e}_k)) + \sum_{m_k=1}^{n_k} \mathbf{p}(\mathbf{n} - m_k \mathbf{e}_k) \mathbf{D}_k(m_k) \right] (-\mathbf{C})^{-1}, \quad \mathbf{n} \in \mathcal{Z}^+, \end{aligned}$$

where $\mathcal{Z}^+ = \mathcal{Z} - \{\mathbf{0}\}$, $\mathbf{q}_k(\mathbf{n}) = \mathbf{0}$ for $\mathbf{n} \notin \mathcal{Z}$ and \mathbf{e}_k ($k \in \mathcal{K}$) denotes the k th unit vector:

$$\mathbf{e}_k = (0, \dots, 0, 1, 0, \dots, 0).$$

k th

Remark IV.1 Throughout the paper, the empty sum is defined as zero.

IV.2 Joint queue length distribution immediately after departures

In this subsection, we consider the vector generating function of the joint queue length distribution immediately after departures of each class. We denote, by $C_k(n; m)$ ($k \in \mathcal{K}$, $n = 1, 2, \dots$, $m = 1, 2, \dots, n$), a randomly chosen class k customer who is a member of a batch of size n and the m th served customer among members of the same batch. Let $N_\nu^{(D_k)}(n; m)$ and $S^{(D_k)}(n; m)$ ($k, \nu \in \mathcal{K}$, $n = 1, 2, \dots$, $m = 1, 2, \dots, n$) denote generic random variables representing the number of class ν customers and the state of the underlying Markov chain, respectively, immediately after the departure of customer $C_k(n; m)$ in steady state. We then define $\mathbf{q}_k^*(\mathbf{z} | n; m)$ ($k \in \mathcal{K}$, $n = 1, 2, \dots$, $m = 1, 2, \dots, n$) as a $1 \times M$ vector whose j th element represents

$$\mathbb{E} \left[\prod_{\nu \in \mathcal{K}} z_\nu^{N_\nu^{(D_k)}(n; m)} 1\{S^{(D_k)}(n; m) = j\} \right],$$

where $1\{\chi\}$ denotes an indicator function of event χ . Because a randomly chosen customer of class k is a member of a batch of size n with probability $n\pi D_k(n)e/\lambda_k$, we have

$$\mathbf{q}_k^*(\mathbf{z}) = \sum_{n=1}^{\infty} \frac{n\pi D_k(n)e}{\lambda_k} \frac{1}{n} \sum_{m=1}^n \mathbf{q}_k^*(\mathbf{z} | n; m). \quad (8)$$

In what follows, we consider $\mathbf{q}_k^*(\mathbf{z} | n; m)$.

We define $\overline{W}_k(n; m)$ ($k \in \mathcal{K}$, $n = 1, 2, \dots$, $m = 1, 2, \dots, n$) as a generic random variable representing the sojourn time of customer $C_k(n; m)$. Note here that

$$\overline{W}_k(n; m) = W_k(n; 1) + H_{k,1} + \dots + H_{k,m},$$

where $W_k(n; 1)$ denotes the actual waiting time of customer $C_k(n; 1)$, and $H_{k,l}$ ($l = 1, 2, \dots, m$) denotes the service time of customer $C_k(n; l)$. By definition, $W_k(n; 1)$ depends only on the past history up to the arrival instant of a batch including customer $C_k(n; m)$. On the other hand, the number of customers in the system immediately after the departure of customer $C_k(n; m)$ is equal to the sum of the $n - m$ customers in the same batch and customers who arrived during the sojourn time of customer $C_k(n; m)$. Note here that the latter is conditionally independent of the past history given the length of the sojourn time and the state of the underlying Markov chain immediately after the arrival of the batch. Thus we have

$$\mathbf{q}_k^*(\mathbf{z} | n; m) = z_k^{n-m} \int_0^\infty d\mathbf{w}_k(x | n; 1) \mathbf{N}^*(x, \mathbf{z}) \left[\int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^m, \quad (9)$$

where

$$\mathbf{N}^*(x, \mathbf{z}) = \exp \left[\left(C + \sum_{k \in \mathcal{K}} D_k^*(z_k) \right) x \right]. \quad (10)$$

Theorem IV.2 *The vector generating function $\mathbf{q}_k^*(\mathbf{z})$ ($k \in \mathcal{K}$) of the joint queue length distribution immediately after departures of class k customers in the steady state is given by*

$$\mathbf{q}_k^*(\mathbf{z}) = \frac{1}{\lambda_k} \sum_{m=1}^{\infty} \sum_{l=0}^{\infty} z_k^l \int_0^{\infty} d\mathbf{v}(x) \mathbf{D}_k(m+l) \mathbf{N}^*(x, \mathbf{z}) \left[\int_0^{\infty} dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^m. \quad (11)$$

Proof. Using (3), (8) and (9), we have

$$\begin{aligned} \mathbf{q}_k^*(\mathbf{z}) &= \sum_{n=1}^{\infty} \frac{\pi \mathbf{D}_k(n) \mathbf{e}}{\lambda_k} \sum_{m=1}^n z_k^{n-m} \int_0^{\infty} \frac{d\mathbf{v}(x) \mathbf{D}_k(n)}{\pi \mathbf{D}_k(n) \mathbf{e}} \mathbf{N}^*(x, \mathbf{z}) \left[\int_0^{\infty} dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^m \\ &= \frac{1}{\lambda_k} \sum_{n=1}^{\infty} \sum_{m=1}^n z_k^{n-m} \int_0^{\infty} d\mathbf{v}(x) \mathbf{D}_k(n) \mathbf{N}^*(x, \mathbf{z}) \left[\int_0^{\infty} dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^m, \end{aligned}$$

from which (11) follows. \square

IV.3 Recursions for discrete phase-type batch sizes

In this subsection, we develop a recursive formula to compute the vector mass function $\mathbf{q}_k(\mathbf{n})$ of the joint queue length immediately after departures of each class under the following assumption.

Assumption IV.1 The batch size distribution of class k is independent of the state of the underlying Markov chain and follows a discrete phase-type distribution with representation $(\boldsymbol{\alpha}_k, \mathbf{P}_k)$, i.e.,

$$\mathbf{D}_k(n) = g_k(n) \mathbf{D}_k, \quad (12)$$

$$g_k(n) = \boldsymbol{\alpha}_k \mathbf{P}_k^{n-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e}, \quad n = 1, 2, \dots, \quad (13)$$

where $\boldsymbol{\alpha}_k$ denotes a $1 \times M_k$ probability vector and \mathbf{P}_k denotes an $M_k \times M_k$ substochastic matrix.

Let $\mathbf{I}(m)$ denote an $m \times m$ identity matrix. When the size of an identity matrix is clear from the context, we suppress (m) .

Lemma IV.1 *Under Assumption IV.1, $\mathbf{q}_k^*(\mathbf{z})$ ($k \in \mathcal{K}$) is given by*

$$\begin{aligned} \mathbf{q}_k^*(\mathbf{z}) &= \frac{1}{\lambda_k} \int_0^{\infty} d\mathbf{v}(x) \mathbf{D}_k \mathbf{N}^*(x, \mathbf{z}) \cdot \left(\boldsymbol{\alpha}_k \otimes \int_0^{\infty} dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right) \\ &\quad \cdot \left[\mathbf{I} - \mathbf{P}_k \otimes \int_0^{\infty} dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^{-1} \\ &\quad \cdot \left[\{ (\mathbf{I} - z_k \mathbf{P}_k)^{-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e} \} \otimes \mathbf{I}(M) \right]. \end{aligned} \quad (14)$$

Proof. Substituting (12) and (13) into (11) and using properties of Kronecker product:

$$\begin{aligned} a\mathbf{A} &= a \otimes \mathbf{A} \text{ for any scalar } a, \text{ and,} \\ (\mathbf{A}_1 \cdots \mathbf{A}_n) &\otimes (\mathbf{B}_1 \cdots \mathbf{B}_n) \\ &= (\mathbf{A}_1 \otimes \mathbf{B}_1) \cdots (\mathbf{A}_n \otimes \mathbf{B}_n) \text{ for any } n = 1, 2, \dots, \end{aligned}$$

we obtain

$$\begin{aligned} \mathbf{q}_k^*(\mathbf{z}) &= \frac{1}{\lambda_k} \int_0^\infty d\mathbf{v}(x) \mathbf{D}_k \mathbf{N}^*(x, \mathbf{z}) \left\{ \boldsymbol{\alpha}_k \sum_{m=1}^\infty \mathbf{P}_k^{m-1} \sum_{l=0}^\infty z_k^l \mathbf{P}_k^l (\mathbf{I} - \mathbf{P}_k) \mathbf{e} \right\} \\ &\quad \otimes \left\{ \left(\int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right)^m \right\} \\ &= \frac{1}{\lambda_k} \int_0^\infty d\mathbf{v}(x) \mathbf{D}_k \mathbf{N}^*(x, \mathbf{z}) \left\{ \boldsymbol{\alpha}_k \sum_{m=1}^\infty \mathbf{P}_k^{m-1} (\mathbf{I} - z_k \mathbf{P}_k)^{-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e} \right\} \\ &\quad \otimes \left\{ \left(\int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right)^m \right\} \\ &= \frac{1}{\lambda_k} \int_0^\infty d\mathbf{v}(x) \mathbf{D}_k \mathbf{N}^*(x, \mathbf{z}) \left(\boldsymbol{\alpha}_k \otimes \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right) \\ &\quad \cdot \sum_{m=1}^\infty \left(\mathbf{P}_k \otimes \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right)^{m-1} \\ &\quad \cdot [\{(\mathbf{I} - z_k \mathbf{P}_k)^{-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e}\} \otimes \mathbf{I}(M)] \\ &= \frac{1}{\lambda_k} \int_0^\infty d\mathbf{v}(x) \mathbf{D}_k \mathbf{N}^*(x, \mathbf{z}) \left(\boldsymbol{\alpha}_k \otimes \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right) \\ &\quad \cdot \left[\mathbf{I} - \mathbf{P}_k \otimes \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^{-1} \\ &\quad \cdot [\{(\mathbf{I} - z_k \mathbf{P}_k)^{-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e}\} \otimes \mathbf{I}(M)], \end{aligned}$$

which completes the proof. \square

We define $\mathbf{v}_k(\mathbf{n})$ ($k \in \mathcal{K}$, $\mathbf{n} \in \mathcal{Z}$) as a $1 \times M$ vector satisfying

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{v}_k(\mathbf{n}) = \int_0^\infty d\mathbf{v}(x) \mathbf{D}_k \mathbf{N}^*(x, \mathbf{z}). \quad (15)$$

We also define $\mathbf{A}_k(\mathbf{n})$ and $\mathbf{\Gamma}_k(\mathbf{n})$ ($k \in \mathcal{K}$, $\mathbf{n} \in \mathcal{Z}$) as $M \times M$ and $MM_k \times MM_k$ matrices satisfying

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{A}_k(\mathbf{n}) = \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}), \quad (16)$$

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{\Gamma}_k(\mathbf{n}) = \left[\mathbf{I} - \mathbf{P}_k \otimes \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right]^{-1}, \quad (17)$$

respectively. Note here that

$$\{(\mathbf{I} - z_k \mathbf{P}_k)^{-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e}\} \otimes \mathbf{I}(M) = \sum_{m=0}^{\infty} z_k^m \{\mathbf{P}_k^m (\mathbf{I} - \mathbf{P}_k) \mathbf{e}\} \otimes \mathbf{I}(M).$$

Thus (14) is rewritten to be

$$\begin{aligned} \mathbf{q}_k^*(\mathbf{z}) &= \frac{1}{\lambda_k} \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \sum_{m=0}^{n_k} \sum_{\substack{\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 \\ = \mathbf{n} - m \mathbf{e}_k}} \mathbf{v}_k(\mathbf{n}_1) \\ &\quad \cdot [\boldsymbol{\alpha}_k \otimes \mathbf{A}_k(\mathbf{n}_2)] \Gamma_k(\mathbf{n}_3) [\{\mathbf{P}_k^m (\mathbf{I} - \mathbf{P}_k) \mathbf{e}\} \otimes \mathbf{I}(M)], \end{aligned} \quad (18)$$

where $\mathbf{n}_j \in \mathcal{Z}$ for $j = 1, 2, 3$. Comparing coefficient vectors of $z_1^{n_1} \cdots z_K^{n_K}$ on both sides of (18), we obtain the following result.

Theorem IV.3 *Under Assumption IV.1, $\mathbf{q}_k(\mathbf{n})$ ($k \in \mathcal{K}$, $\mathbf{n} \in \mathcal{Z}$) is given by*

$$\mathbf{q}_k(\mathbf{n}) = \frac{1}{\lambda_k} \sum_{m=0}^{n_k} \sum_{\substack{\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 \\ = \mathbf{n} - m \mathbf{e}_k}} \mathbf{v}_k(\mathbf{n}_1) [\boldsymbol{\alpha}_k \otimes \mathbf{A}_k(\mathbf{n}_2)] \Gamma_k(\mathbf{n}_3) [\mathbf{P}_k^m (\mathbf{I} - \mathbf{P}_k) \mathbf{e} \otimes \mathbf{I}(M)],$$

where $\mathbf{n}_j \in \mathcal{Z}$ for $j = 1, 2, 3$.

Theorem IV.3 implies that the computation of $\mathbf{q}_k(\mathbf{n})$ is reduced to those of $\mathbf{v}_k(\mathbf{n})$, $\mathbf{A}_k(\mathbf{n})$ and $\Gamma_k(\mathbf{n})$, which are discussed in the rest of this subsection.

We first consider the $\mathbf{A}_k(\mathbf{n})$. Let θ denote the maximum absolute value of diagonal elements of \mathbf{C} . We define $\mathbf{F}_m(\mathbf{n})$ ($m = 0, 1, \dots$, $\mathbf{n} \in \mathcal{Z}$) as an $M \times M$ matrix that satisfies

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{F}_m(\mathbf{n}) = \left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z_k) \right) \right]^m. \quad (19)$$

Lemma IV.2 $\mathbf{A}_k(\mathbf{n})$ is given by

$$\mathbf{A}_k(\mathbf{n}) = \sum_{m=0}^{\infty} \gamma_k^{(m)}(\theta) \mathbf{F}_m(\mathbf{n}), \quad k \in \mathcal{K}, \mathbf{n} \in \mathcal{Z}, \quad (20)$$

where

$$\gamma_k^{(m)}(\theta) = \int_0^{\infty} e^{-\theta y} \frac{(\theta y)^m}{m!} dH_k(y), \quad k \in \mathcal{K}, m = 0, 1, \dots, \quad (21)$$

and $\mathbf{F}_m(\mathbf{n})$'s are recursively determined by

$$\mathbf{F}_0(\mathbf{n}) = \begin{cases} \mathbf{I}, & \text{if } \mathbf{n} = \mathbf{0}, \\ \mathbf{O}, & \text{otherwise,} \end{cases} \quad (22)$$

and for $m = 0, 1, \dots$,

$$\mathbf{F}_{m+1}(\mathbf{n}) = \mathbf{F}_m(\mathbf{n})(\mathbf{I} + \theta^{-1}\mathbf{C}) + \theta^{-1} \sum_{k \in \mathcal{K}} \sum_{l_k=1}^{n_k} \mathbf{F}_m(\mathbf{n} - l_k \mathbf{e}_k) \mathbf{D}_k(l_k), \quad \mathbf{n} \in \mathcal{Z}. \quad (23)$$

Proof. From (10), (16) and (21), we obtain

$$\begin{aligned} \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{A}_k(\mathbf{n}) &= \int_0^\infty dH_k(y) \exp \left[\left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z_k) \right) y \right] \\ &= \sum_{m=0}^\infty \int_0^\infty e^{-\theta y} \frac{(\theta y)^m}{m!} dH_k(y) \left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z_k) \right) \right]^m \\ &= \sum_{m=0}^\infty \gamma_k^{(m)}(\theta) \left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z_k) \right) \right]^m. \end{aligned} \quad (24)$$

Substituting (19) into (24) and changing the order of summations, we have

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{A}_k(\mathbf{n}) = \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \sum_{m=0}^\infty \gamma_k^{(m)}(\theta) \mathbf{F}_m(\mathbf{n}). \quad (25)$$

Comparing the coefficient matrices of $z_1^{n_1} \cdots z_K^{n_K}$ on both sides of (25), we obtain (20). (22) is clear from the definition. The remaining is to show (23). From (6) and (19), we have for $m = 0, 1, \dots$,

$$\begin{aligned} &\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{F}_{m+1}(\mathbf{n}) \\ &= \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{F}_m(\mathbf{n}) \left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z_k) \right) \right] \\ &= \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{F}_m(\mathbf{n}) (\mathbf{I} + \theta^{-1}\mathbf{C}) \\ &\quad + \sum_{\mathbf{n} \in \mathcal{Z}} \sum_{k \in \mathcal{K}} \left[\sum_{l_k=1}^\infty z_1^{n_1} \cdots z_{k-1}^{n_{k-1}} z_k^{n_k+l_k} z_{k+1}^{n_{k+1}} \cdots z_K^{n_K} \mathbf{F}_m(\mathbf{n}) \cdot \theta^{-1} \mathbf{D}_k(l_k) \right] \\ &= \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \mathbf{F}_m(\mathbf{n}) (\mathbf{I} + \theta^{-1}\mathbf{C}) \\ &\quad + \sum_{\mathbf{n} \in \mathcal{Z}^+} z_1^{n_1} \cdots z_K^{n_K} \theta^{-1} \sum_{k \in \mathcal{K}} \sum_{l_k=1}^{n_k} \mathbf{F}_m(\mathbf{n} - l_k \mathbf{e}_k) \mathbf{D}_k(l_k). \end{aligned}$$

Comparing the coefficient vectors of $z_1^{n_1} \cdots z_K^{n_K}$ on both sides of the above equation, we obtain (23). \square

Next we consider the $\Gamma_k(\mathbf{n})$ in (17).

Lemma IV.3 $\Gamma_k(\mathbf{n})$ ($k \in \mathcal{K}$, $\mathbf{n} \in \mathcal{Z}$) is determined by the following recursion:

$$\begin{aligned}\Gamma_k(\mathbf{0}) &= [\mathbf{I} - \mathbf{P}_k \otimes \mathbf{A}_k(\mathbf{0})]^{-1}, \\ \Gamma_k(\mathbf{n}) &= \sum_{\substack{0 \leq l \leq \mathbf{n} \\ l \neq \mathbf{0}}} \Gamma_k(\mathbf{n} - l) [\mathbf{P}_k \otimes \mathbf{A}_k(l)] \Gamma_k(\mathbf{0}), \quad \mathbf{n} \in \mathcal{Z}^+.\end{aligned}$$

Proof. Note first that (17) is equivalent to

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \Gamma_k(\mathbf{n}) \left[\mathbf{I} - \mathbf{P}_k \otimes \int_0^\infty dH_k(y) \mathbf{N}^*(y, \mathbf{z}) \right] = \mathbf{I}.$$

Substituting (16) into the above equation, we have

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \Gamma_k(\mathbf{n}) \left[\mathbf{I} - \mathbf{P}_k \otimes \sum_{l \in \mathcal{Z}} z_1^{l_1} \cdots z_K^{l_K} \mathbf{A}_k(l) \right] = \mathbf{I},$$

from which it follows that

$$\sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \Gamma_k(\mathbf{n}) - \sum_{\mathbf{n} \in \mathcal{Z}} z_1^{n_1} \cdots z_K^{n_K} \sum_{\mathbf{0} \leq l \leq \mathbf{n}} \Gamma_k(\mathbf{n} - l) [\mathbf{P}_k \otimes \mathbf{A}_k(l)] = \mathbf{I}.$$

Comparing the coefficient matrices of $z_1^{n_1} \cdots z_K^{n_K}$ on both sides of the above equation, we have

$$\begin{aligned}\Gamma_k(\mathbf{0}) - \Gamma_k(\mathbf{0}) [\mathbf{P}_k \otimes \mathbf{A}_k(\mathbf{0})] &= \mathbf{I}, \\ \Gamma_k(\mathbf{n}) - \sum_{\mathbf{0} \leq l \leq \mathbf{n}} \Gamma_k(\mathbf{n} - l) [\mathbf{P}_k \otimes \mathbf{A}_k(l)] &= \mathbf{O}, \quad \mathbf{n} \in \mathcal{Z}^+, \end{aligned}$$

or equivalently,

$$\Gamma_k(\mathbf{0}) = [\mathbf{I} - \mathbf{P}_k \otimes \mathbf{A}_k(\mathbf{0})]^{-1},$$

and for $\mathbf{n} \in \mathcal{Z}^+$,

$$\Gamma_k(\mathbf{n}) = \sum_{\substack{0 \leq l \leq \mathbf{n} \\ l \neq \mathbf{0}}} \Gamma_k(\mathbf{n} - l) [\mathbf{P}_k \otimes \mathbf{A}_k(l)] [\mathbf{I} - \mathbf{P}_k \otimes \mathbf{A}_k(\mathbf{0})]^{-1},$$

from which Lemma IV.3 follows. \square

Finally, we consider the $\mathbf{v}_k(\mathbf{n})$ in (15). In a very similar way to derive (20), we obtain the following lemma.

Lemma IV.4 $\mathbf{v}_k(\mathbf{n})$ ($k \in \mathcal{K}$, $\mathbf{n} \in \mathcal{Z}$) is given by

$$\mathbf{v}_k(\mathbf{n}) = \sum_{m=0}^{\infty} \mathbf{v}^{(m)}(\theta) \mathbf{D}_k \mathbf{F}_m(\mathbf{n}),$$

where $\mathbf{F}_m(\mathbf{n})$ is given in (22) and (23), and

$$\mathbf{v}^{(m)}(\theta) = \int_0^\infty e^{-\theta x} \frac{(\theta x)^m}{m!} d\mathbf{v}(x), \quad m = 0, 1, \dots$$

Thus $\mathbf{v}_k(\mathbf{n})$ is given in terms of the $\mathbf{v}^{(m)}(\theta)$. Because the computation of the $\mathbf{v}^{(m)}(\theta)$ has already been studied in [17], we summarize the result below. As for the details, readers are referred to Lemma 3 in [17].

Note first that

$$\sum_{m=0}^{\infty} z^m \mathbf{v}^{(m)}(\theta) = \mathbf{v}^*(\theta - \theta z), \quad (26)$$

where $\mathbf{v}^*(s)$ is given in (1). Thus, substituting $\theta - \theta z$ for s in (1) and using (26) yield

$$\sum_{m=0}^{\infty} z^m \mathbf{v}^{(m)}(\theta) \left[(\theta - \theta z) \mathbf{I} + \mathbf{C} + \sum_{m=0}^{\infty} z^m \mathbf{D}^{(m)}(\theta) \right] = (\theta - \theta z)(1 - \rho) \boldsymbol{\kappa}, \quad (27)$$

where $\mathbf{D}^{(m)}(\theta)$ denotes

$$\mathbf{D}^{(m)}(\theta) = \int_0^{\infty} e^{-\theta x} \frac{(\theta x)^m}{m!} d\mathbf{D}(x).$$

Comparing the coefficient vectors of z^m ($m = 0, 1, \dots$) on both sides of (27), we can show that the $\mathbf{v}^{(m)}(\theta)$ is identical to the steady-state solution of a Markov chain of M/G/1 type whose transition probability matrix is given by [17]

$$\begin{bmatrix} B_0 + B_1 & B_2 & B_3 & B_4 & \cdots \\ B_0 & B_1 & B_2 & B_3 & \cdots \\ O & B_0 & B_1 & B_2 & \cdots \\ O & O & B_0 & B_1 & \cdots \\ O & O & O & B_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

where

$$B_0 = \mathbf{I} + \theta^{-1}(\mathbf{C} + \mathbf{D}^{(0)}(\theta)), \quad B_m = \theta^{-1} \mathbf{D}^{(m)}(\theta), \quad m \geq 1.$$

Thus applying the general theory of Markov chains of M/G/1 type [9], we can compute the $\mathbf{v}^{(m)}(\theta)$. As for the truncation and stopping criteria in computing the steady-state solution of Markov chains of M/G/1 type, readers are referred to [9, 12].

Let $\mathbf{d}_k^{(m)}(\theta)$ ($k \in \mathcal{K}$, $m = 0, 1, \dots$) denote a $1 \times M_k$ vector which satisfies

$$\sum_{m=0}^{\infty} z^m \mathbf{d}_k^{(m)}(\theta) = H_k^*(\theta - \theta z) \boldsymbol{\alpha}_k (\mathbf{I} - \mathbf{P}_k) [\mathbf{I} - H_k^*(\theta - \theta z) \mathbf{P}_k]^{-1}. \quad (28)$$

Lemma IV.5 *Under Assumption IV.1, $\mathbf{D}^{(m)}(\theta)$ is given by*

$$\mathbf{D}^{(m)}(\theta) = \sum_{k \in \mathcal{K}} \mathbf{d}_k^{(m)}(\theta) \mathbf{e} \mathbf{D}_k, \quad m = 0, 1, \dots,$$

where $\mathbf{d}_k^{(m)}(\theta)$'s ($k \in \mathcal{K}$) are recursively determined by

$$\mathbf{d}_k^{(0)}(\theta) = \gamma_k^{(0)}(\theta) \boldsymbol{\alpha}_k (\mathbf{I} - \mathbf{P}_k) \left[\mathbf{I} - \gamma_k^{(0)}(\theta) \mathbf{P}_k \right]^{-1},$$

and for $m = 1, 2, \dots$,

$$\mathbf{d}_k^{(m)}(\theta) = \frac{\gamma_k^{(m)}(\theta)}{\gamma_k^{(0)}(\theta)} \mathbf{d}_k^{(0)}(\theta) + \left[\sum_{l=1}^m \gamma_k^{(l)}(\theta) \mathbf{d}_k^{(m-l)}(\theta) \right] \mathbf{P}_k \left[\mathbf{I} - \gamma_k^{(0)}(\theta) \mathbf{P}_k \right]^{-1}.$$

Proof. Note first that

$$\sum_{m=0}^{\infty} z^m \mathbf{D}^{(m)}(\theta) = \mathbf{D}^*(\theta - \theta z),$$

where $\mathbf{D}^*(s)$ is given in (2). Thus, substituting $\theta - \theta z$ for s in (2) and using (12) and (13), we have

$$\begin{aligned} \sum_{m=0}^{\infty} z^m \mathbf{D}^{(m)}(\theta) &= \sum_{k \in \mathcal{K}} \sum_{n=1}^{\infty} \boldsymbol{\alpha}_k \mathbf{P}_k^{n-1} (\mathbf{I} - \mathbf{P}_k) \mathbf{e} \{H_k^*(\theta - \theta z)\}^n \mathbf{D}_k \\ &= \sum_{k \in \mathcal{K}} H_k^*(\theta - \theta z) \boldsymbol{\alpha}_k (\mathbf{I} - \mathbf{P}_k) [\mathbf{I} - H_k^*(\theta - \theta z) \mathbf{P}_k]^{-1} \mathbf{e} \mathbf{D}_k. \end{aligned} \quad (29)$$

It then follows from (28) and (29) that

$$\begin{aligned} \sum_{m=0}^{\infty} z^m \mathbf{D}^{(m)}(\theta) &= \sum_{k \in \mathcal{K}} \sum_{m=0}^{\infty} z^m \mathbf{d}_k^{(m)}(\theta) \mathbf{e} \mathbf{D}_k \\ &= \sum_{m=0}^{\infty} z^m \sum_{k \in \mathcal{K}} \mathbf{d}_k^{(m)}(\theta) \mathbf{e} \mathbf{D}_k. \end{aligned} \quad (30)$$

Note here that

$$H_k^*(\theta - \theta z) = \sum_{m=0}^{\infty} z^m \gamma_k^{(m)}(\theta), \quad k \in \mathcal{K}. \quad (31)$$

Thus from (28) and (31), we have

$$\sum_{m=0}^{\infty} z^m \mathbf{d}_k^{(m)}(\theta) \left[\mathbf{I} - \sum_{l=0}^{\infty} z^l \gamma_k^{(l)}(\theta) \mathbf{P}_k \right] = \sum_{m=0}^{\infty} z^m \gamma_k^{(m)}(\theta) \boldsymbol{\alpha}_k [\mathbf{I} - \mathbf{P}_k],$$

or equivalently,

$$\sum_{m=0}^{\infty} z^m \mathbf{d}_k^{(m)}(\theta) - \sum_{m=0}^{\infty} z^m \sum_{l=0}^m \mathbf{d}_k^{(m-l)}(\theta) \gamma_k^{(l)}(\theta) \mathbf{P}_k = \sum_{m=0}^{\infty} z^m \gamma_k^{(m)}(\theta) \boldsymbol{\alpha}_k [\mathbf{I} - \mathbf{P}_k].$$

Comparing the coefficient vectors of z^m ($m = 0, 1, \dots$) on both sides of the above equation, we have

$$\mathbf{d}_k^{(0)}(\theta) [\mathbf{I} - \gamma_k^{(0)}(\theta) \mathbf{P}_k] = \gamma_k^{(0)}(\theta) \boldsymbol{\alpha}_k [\mathbf{I} - \mathbf{P}_k], \quad (32)$$

and for $m = 1, 2, \dots$,

$$\mathbf{d}_k^{(m)}(\theta) \left[\mathbf{I} - \gamma_k^{(0)}(\theta) \mathbf{P}_k \right] - \sum_{l=1}^m \mathbf{d}_k^{(m-l)}(\theta) \gamma_k^{(l)}(\theta) \mathbf{P}_k = \gamma_k^{(m)}(\theta) \boldsymbol{\alpha}_k [\mathbf{I} - \mathbf{P}_k]. \quad (33)$$

Lemma IV.5 now follows from (30), (32) and (33). \square

V Implementations of Recursions

In this section, we consider the implementation of recursions for $\mathbf{A}_k(\mathbf{n})$, $\mathbf{v}_k(\mathbf{n})$ and $\Gamma_k(\mathbf{n})$, derived in the preceding section. At a glance, they would seem to be easy to implement. Contrary to the single arrival case [16, 17], however, the computation of the $\mathbf{F}_m(\mathbf{n})$ appeared in $\mathbf{A}_k(\mathbf{n})$ and $\mathbf{v}_k(\mathbf{n})$ is not straightforward, because the direct implementation of the recursion requires very huge memory space and time-consuming. In what follows, we construct a numerically feasible procedure to compute the approximate sequences of $\mathbf{A}_k(\mathbf{n})$ and $\mathbf{v}_k(\mathbf{n})$, avoiding the computation of $\mathbf{F}_m(\mathbf{n})$'s whose contributions to $\mathbf{A}_k(\mathbf{n})$ and $\mathbf{v}_k(\mathbf{n})$ are negligible, and establish the truncation/stopping criteria and error bounds. Further, we propose a computational procedure for the $\Gamma_k(\mathbf{n})$ and establish the error bound.

We start with $\mathbf{A}_k(\mathbf{n})$ and $\mathbf{v}_k(\mathbf{n})$. Note first that for $k \in \mathcal{K}$,

$$\sum_{\mathbf{n} \in \mathcal{Z}} \mathbf{A}_k(\mathbf{n}) \mathbf{e} = \mathbf{e}, \quad \sum_{\mathbf{n} \in \mathcal{Z}} \mathbf{v}_k(\mathbf{n}) \mathbf{e} = \lambda_k^{(\text{B})},$$

where

$$\lambda_k^{(\text{B})} = \boldsymbol{\pi} \mathbf{D}_k \mathbf{e}.$$

In numerical computation, we have to stop the computation of those sequences. Thus we develop a numerical procedure to obtain approximations $\check{\mathbf{A}}_k(\mathbf{n})$ and $\check{\mathbf{v}}_k(\mathbf{n})$ to $\mathbf{A}_k(\mathbf{n})$ and $\mathbf{v}_k(\mathbf{n})$, respectively, while ensuring the following error bounds: For a given ε ($0 < \varepsilon < 1$), there exist $n_A(k)$ and $n_v(k)$ such that

$$\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_A(k)}} \check{\mathbf{A}}_k(\mathbf{n}) \mathbf{e} > (1 - \varepsilon) \mathbf{e}, \quad (34)$$

$$\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_v(k)}} \check{\mathbf{v}}_k(\mathbf{n}) \mathbf{e} > (1 - \varepsilon) \lambda_k^{(\text{B})}, \quad (35)$$

where $|\mathbf{n}| = \sum_{k \in \mathcal{K}} n_k$ for $\mathbf{n} \in \mathcal{Z}$. In what follows, we first show our proposed algorithm and then show that the above error bounds are satisfied.

Numerical algorithm for $A_k(n)$ and $v_k(n)$

Input.

Stopping criterion : ε ($0 < \varepsilon < 1$),

Underlying Markov chain : C, D_k ($k \in \mathcal{K}$),

Batch size distribution : α_k, P_k ($k \in \mathcal{K}$),

Service time distribution : $H_k(x)$ ($k \in \mathcal{K}$).

Step 1. Choose ε_F ($0 < \varepsilon_F < 1$) such that

$$\frac{\varepsilon_F}{\varepsilon} < \min_{k \in \mathcal{K}} \min \left(\frac{1}{\theta h_k}, \frac{\lambda_k^{(B)}}{\theta \bar{v}^{(1)} D_k e} \right), \quad (36)$$

where $\bar{v}^{(1)} = -\lim_{s \rightarrow 0+} d v^*(s)/ds$, whose computational procedure can be found in [11]. Then compute the $\gamma_k^{(m)}(\theta)$ and the $v^{(m)}(\theta)$ until they satisfy

$$\sum_{m=0}^{m_\gamma(k)} \gamma_k^{(m)}(\theta) (1 - \varepsilon_F)^m > 1 - \varepsilon, \quad k \in \mathcal{K}, \quad (37)$$

$$\sum_{m=0}^{m_v(k)} v^{(m)}(\theta) D_k e (1 - \varepsilon_F)^m > (1 - \varepsilon) \lambda_k^{(B)}, \quad k \in \mathcal{K}, \quad (38)$$

for some $m_\gamma(k)$ and $m_v(k)$, respectively. Define m_{\max} as

$$m_{\max} = \max_{k \in \mathcal{K}} \max(m_\gamma(k), m_v(k)).$$

Step 2. Choose ε_g such that $0 < \varepsilon_g < \varepsilon_F$. Then compute $g_k(n)$ ($n = 1, 2, \dots$) by (13) until the $g_k(n)$ satisfies

$$\theta^{-1} \sum_{n=1}^{n_g(k)} g_k(n) D_k e > \theta^{-1} D_k e - \frac{\varepsilon_g}{K} e, \quad k \in \mathcal{K}, \quad (39)$$

for some $n_g(k)$.

Step 3. Compute $\check{A}_k(n)$ and $\check{v}_k(n)$ by the following procedure, where the initial values of $\check{A}_k(n)$ and $\check{v}_k(n)$ ($n \in \mathcal{Z}$) are assumed to be O and 0 , respectively.

Step (3-a). Set $\check{F}_0(0) = I$ and $n_F^{(0)} = 0$. Also set $\check{A}_k(0) = \gamma_k^{(0)}(\theta) I$ and $\check{v}_k(0) = v^{(0)}(\theta) D_k$ for all $k \in \mathcal{K}$.

Step (3-b). Set $n_F^{(1)} = \max_{k \in \mathcal{K}} n_g(k)$ and $m = 1$, and compute $\check{F}_1(n)$'s ($|n| \leq n_F^{(1)}$) by

$$\check{F}_1(n) = \begin{cases} I + \theta^{-1} C, & \text{if } n = 0, \\ \theta^{-1} g_k(n_k) D_k, & \text{if } n \in \mathcal{Z}_k(F_1), k \in \mathcal{K}, \\ O, & \text{otherwise,} \end{cases} \quad (40)$$

where

$$\mathcal{Z}_k(F_1) = \{\mathbf{n}; \mathbf{n} = n_k \mathbf{e}_k, n_k = 1, 2, \dots, n_g(k)\}, \quad k \in \mathcal{K}.$$

Step (3-c). For each $k \in \mathcal{K}$, if $m \leq m_\gamma(k)$, add $\gamma_k^{(m)}(\theta) \check{\mathbf{F}}_m(\mathbf{n})$ to $\check{\mathbf{A}}_k(\mathbf{n})$ for all \mathbf{n} ($|\mathbf{n}| \leq n_F^{(m)}$). Also, for each $k \in \mathcal{K}$, if $m \leq m_v(k)$, add $\mathbf{v}^{(m)}(\theta) \mathbf{D}_k \check{\mathbf{F}}_m(\mathbf{n})$ to $\check{\mathbf{v}}_k(\mathbf{n})$ for all \mathbf{n} ($|\mathbf{n}| \leq n_F^{(m)}$).

Step (3-d). If $m \geq m_{\max}$, stop computing, and otherwise, add one to m and go to Step (3-e).

Step (3-e). For each $n = 0, 1, \dots$, compute $\check{\mathbf{F}}_m(\mathbf{n})$'s ($|\mathbf{n}| = n$) by

$$\begin{aligned} \check{\mathbf{F}}_m(\mathbf{n}) = & U\left(n_F^{(m-1)} - |\mathbf{n}|\right) \check{\mathbf{F}}_{m-1}(\mathbf{n})(\mathbf{I} + \theta^{-1} \mathbf{C}) \\ & + \theta^{-1} \sum_{k \in \mathcal{K}} \sum_{l_k=1}^{\min(n_k, n_g(k))} U\left(n_F^{(m-1)} - |\mathbf{n} - l_k \mathbf{e}_k|\right) \\ & \cdot \check{\mathbf{F}}_{m-1}(\mathbf{n} - l_k \mathbf{e}_k) g_k(l_k) \mathbf{D}_k, \end{aligned} \quad (41)$$

until $\check{\mathbf{F}}_m(\mathbf{n})$'s satisfy $\sum_{|\mathbf{n}| \leq n^*} \check{\mathbf{F}}_m(\mathbf{n}) \mathbf{e} > (1 - \varepsilon_F)^m \mathbf{e}$ for some n^* , where $U(x)$ denotes a unit step function:

$$U(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

Let $n_F^{(m)} = n^*$ and go to Step (3-c).

Remark V.1 Note that $\check{\mathbf{A}}_k(\mathbf{n})$ ($|\mathbf{n}| \leq n_A(k)$) and $\check{\mathbf{v}}_k(\mathbf{n})$ ($|\mathbf{n}| \leq n_v(k)$) obtained by the above algorithm satisfy

$$\begin{aligned} \check{\mathbf{A}}_k(\mathbf{n}) &= \sum_{m=0}^{m_\gamma(k)} U\left(n_F^{(m)} - |\mathbf{n}|\right) \gamma_k^{(m)}(\theta) \check{\mathbf{F}}_m(\mathbf{n}), \\ \check{\mathbf{v}}_k(\mathbf{n}) &= \sum_{m=0}^{m_v(k)} U\left(n_F^{(m)} - |\mathbf{n}|\right) \mathbf{v}^{(m)}(\theta) \mathbf{D}_k \check{\mathbf{F}}_m(\mathbf{n}), \end{aligned}$$

respectively, where

$$\begin{aligned} n_A(k) &= \max\left(n_F^{(m)}; m = 0, 1, \dots, m_\gamma(k)\right), \\ n_v(k) &= \max\left(n_F^{(m)}; m = 0, 1, \dots, m_v(k)\right). \end{aligned} \quad (42)$$

Remark V.2 If we are interested only in the $\mathbf{p}(\mathbf{n})$ ($|\mathbf{n}| \leq N_p$) for some N_p , we do not need to compute $\check{\mathbf{F}}_m(\mathbf{n})$ for \mathbf{n} such that $|\mathbf{n}| > N_p$. Thus, in this case, $n_g(k)$ is redefined as $\min(n_g(k), N_p)$ and Step (3-e) is replaced by

Step (3–e’). For each n ($n = 0, 1, \dots$), compute $\check{\mathbf{F}}_m(\mathbf{n})$ ’s ($|\mathbf{n}| = n$, $\mathbf{n} \in \mathcal{Z}$) by (41) until $\check{\mathbf{F}}_m(\mathbf{n})$ ’s satisfy

$$\sum_{|\mathbf{n}| \leq n^*} \check{\mathbf{F}}_m(\mathbf{n}) \mathbf{e} > (1 - \varepsilon_F)^m \mathbf{e},$$

for some n^* , or $n = N_p$, whichever occurs first. Let $n_F^{(m)} = \min(n^*, N_p)$ and go to Step (3–c).

This procedure can save the computational cost, while maintaining the accuracy of the results.

We now provide two lemmas that ensure the above procedure eventually stops.

Lemma V.1 *There exist integers $m_\gamma(k)$ and $m_v(k)$ satisfying (37) and (38), respectively.*

Proof. Substituting $1 - \varepsilon_F$ for z in (31), we have

$$\sum_{m=0}^{\infty} \gamma_k^{(m)}(\theta) (1 - \varepsilon_F)^m = H_k^*(\theta \varepsilon_F), \quad (43)$$

where $H_k^*(s)$ denotes the LST of $H_k(x)$. Similarly, from (26), we have

$$\sum_{m=0}^{\infty} \mathbf{v}^{(m)}(\theta) \mathbf{D}_k \mathbf{e} (1 - \varepsilon_F)^m = \mathbf{v}^*(\theta \varepsilon_F) \mathbf{D}_k \mathbf{e}. \quad (44)$$

Note here that

$$H_k^*(\theta \varepsilon_F) > 1 - h_k \cdot (\theta \varepsilon_F), \quad k \in \mathcal{K}, \quad (45)$$

$$\mathbf{v}^*(\theta \varepsilon_F) \mathbf{D}_k \mathbf{e} > \lambda_k^{(B)} - \bar{\mathbf{v}}^{(1)} \mathbf{D}_k \mathbf{e} \cdot (\theta \varepsilon_F), \quad k \in \mathcal{K}, \quad (46)$$

because $H_k^*(s)$ and each element of $\mathbf{v}^*(s)$ are convex functions of s . Note also that (36) is equivalent to

$$1 - h_k \theta \varepsilon_F > 1 - \varepsilon, \quad k \in \mathcal{K}, \quad (47)$$

$$\lambda_k^{(B)} - \bar{\mathbf{v}}^{(1)} \mathbf{D}_k \mathbf{e} \theta \varepsilon_F > (1 - \varepsilon) \lambda_k^{(B)}, \quad k \in \mathcal{K}. \quad (48)$$

It then follows from (43)–(48) that

$$\begin{aligned} \sum_{m=0}^{\infty} \gamma_k^{(m)}(\theta) (1 - \varepsilon_F)^m &> 1 - h_k \theta \varepsilon_F > 1 - \varepsilon, \\ \sum_{m=0}^{\infty} \mathbf{v}^{(m)}(\theta) \mathbf{D}_k \mathbf{e} (1 - \varepsilon_F)^m &> \lambda_k^{(B)} - \bar{\mathbf{v}}^{(1)} \mathbf{D}_k \mathbf{e} \theta \varepsilon_F > (1 - \varepsilon) \lambda_k^{(B)}, \end{aligned}$$

which complete the proof. \square

Lemma V.2 *There exists an integer $n_F^{(m)}$ such that*

$$\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(m)}}} \check{\mathbf{F}}_m(\mathbf{n})\mathbf{e} > (1 - \varepsilon_F)^m \mathbf{e}, \quad \forall m = 1, 2, \dots, m_{\max}. \quad (49)$$

Proof. We first consider the case $m = 1$. It follows from (39) and (40) that

$$\begin{aligned} \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(1)}}} \check{\mathbf{F}}_1(\mathbf{n})\mathbf{e} &= \left[\mathbf{I} + \theta^{-1}\mathbf{C} + \theta^{-1} \sum_{k \in \mathcal{K}} \sum_{n_k=1}^{n_g(k)} g_k(n_k) \mathbf{D}_k \right] \mathbf{e} \\ &> \mathbf{e} + \theta^{-1}\mathbf{C}\mathbf{e} + \sum_{k \in \mathcal{K}} \left[\theta^{-1} \mathbf{D}_k \mathbf{e} - \frac{\varepsilon_g}{K} \mathbf{e} \right] \\ &= (1 - \varepsilon_g) \mathbf{e} > (1 - \varepsilon_F) \mathbf{e}, \end{aligned} \quad (50)$$

where we use $(\mathbf{C} + \mathbf{D})\mathbf{e} = \mathbf{0}$ and $\varepsilon_g < \varepsilon_F$.

Suppose that for some m ($1 \leq m \leq m_{\max} - 1$), there exists an integer $n_F^{(m)}$ such that

$$\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(m)}}} \check{\mathbf{F}}_m(\mathbf{n})\mathbf{e} > (1 - \varepsilon_F)^m \mathbf{e}. \quad (51)$$

Using (40) and (41), we have

$$\begin{aligned} \check{\mathbf{F}}_{m+1}(\mathbf{n}) &= U \left(n_F^{(m)} - |\mathbf{n}| \right) \check{\mathbf{F}}_m(\mathbf{n}) \check{\mathbf{F}}_1(\mathbf{0}) \\ &\quad + \sum_{k \in \mathcal{K}} \sum_{l_k=1}^{\min(n_k, n_g(k))} U \left(n_F^{(m)} - |\mathbf{n} - l_k \mathbf{e}_k| \right) \\ &\quad \cdot \check{\mathbf{F}}_m(\mathbf{n} - l_k \mathbf{e}_k) \check{\mathbf{F}}_1(l_k \mathbf{e}_k), \quad \mathbf{n} \in \mathcal{Z}. \end{aligned} \quad (52)$$

It then follows from (50), (51) and (52) that

$$\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(m)} + n_F^{(1)}}} \check{\mathbf{F}}_{m+1}(\mathbf{n})\mathbf{e} = \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(m)}}} \check{\mathbf{F}}_m(\mathbf{n}) \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(1)}}} \check{\mathbf{F}}_1(\mathbf{n})\mathbf{e} > (1 - \varepsilon_F)^{m+1} \mathbf{e}. \quad (53)$$

Thus we can choose $n_F^{(m+1)}$ in such a way that

$$n_F^{(m+1)} \leq n_F^{(m)} + n_F^{(1)}, \quad m = 1, 2, \dots, m_{\max} - 1,$$

which completes the proof. \square

Theorem V.1 *For $0 < \varepsilon < 1$, the $\check{\mathbf{A}}_k(\mathbf{n})$ and the $\check{\mathbf{v}}_k(\mathbf{n})$, computed by Step 3, satisfy error bounds (34) and (35), respectively.*

Proof. Using Lemma V.1, Lemma V.2 and (42), we obtain

$$\begin{aligned}
\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_A(k)}} \check{\mathbf{A}}_k(\mathbf{n})\mathbf{e} &= \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_A(k)}} \sum_{m=0}^{m_\gamma(k)} U\left(n_F^{(m)} - |\mathbf{n}|\right) \gamma_k^{(m)}(\theta) \check{\mathbf{F}}_m(\mathbf{n})\mathbf{e} \\
&= \sum_{m=0}^{m_\gamma(k)} \gamma_k^{(m)}(\theta) \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_F^{(m)}}} \check{\mathbf{F}}_m(\mathbf{n})\mathbf{e} \\
&> \sum_{m=0}^{m_\gamma(k)} \gamma_k^{(m)}(\theta) (1 - \varepsilon_F)^m \mathbf{e} > (1 - \varepsilon) \mathbf{e}, \quad k \in \mathcal{K}.
\end{aligned}$$

In the same way, we can obtain (35), so that the proof for the $\check{\mathbf{v}}_k(\mathbf{n})$ is omitted. \square

Finally, we consider the $\Gamma_k(\mathbf{n})$. Note here that

$$\sum_{\mathbf{n} \in \mathcal{Z}} \Gamma_k(\mathbf{n})\mathbf{e} = \{(\mathbf{I} - \mathbf{P}_k)^{-1} \mathbf{e}(M_k)\} \otimes \mathbf{e}(M),$$

where $\mathbf{e}(m)$ denotes an $m \times 1$ vector whose elements are all equal to one. Keeping the above equation in mind, we propose to compute an approximation $\check{\Gamma}_k(\mathbf{n})$ to $\Gamma_k(\mathbf{n})$ in the following way.

Step 4. For each $k \in \mathcal{K}$, compute $\check{\Gamma}_k(\mathbf{n})$'s ($|\mathbf{n}| = n$) for $n = 0, 1, \dots$ by

$$\check{\Gamma}_k(\mathbf{0}) = \left[\mathbf{I} - \mathbf{P}_k \otimes \check{\mathbf{A}}_k(\mathbf{0}) \right]^{-1}, \quad (54)$$

$$\check{\Gamma}_k(\mathbf{n}) = \sum_{\substack{0 \leq l \leq n \\ l \neq 0}} U(n_A(k) - |l|) \check{\Gamma}_k(\mathbf{n} - l) \left[\mathbf{P}_k \otimes \check{\mathbf{A}}_k(l) \right] \check{\Gamma}_k(\mathbf{0}), \quad \mathbf{n} \in \mathcal{Z}^+, \quad (55)$$

until $\check{\Gamma}_k(\mathbf{n})$'s satisfy

$$\begin{aligned}
\sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_\Gamma(k)}} \check{\Gamma}_k(\mathbf{n})\mathbf{e} &> \{(\mathbf{I} - \mathbf{P}_k)^{-1} \mathbf{e}(M_k)\} \otimes \mathbf{e}(M) \\
&\quad - \varepsilon \{(\mathbf{I} - \mathbf{P}_k)^{-2} \mathbf{P}_k \mathbf{e}(M_k)\} \otimes \mathbf{e}(M), \quad (56)
\end{aligned}$$

for some integer $n_\Gamma(k)$.

Remark V.3 Let G_k ($k \in \mathcal{K}$) denote a generic random variable representing a batch size of class k . We then have

$$(\boldsymbol{\alpha}_k \otimes \boldsymbol{\pi}) \sum_{\mathbf{n} \in \mathcal{Z}} \Gamma_k(\mathbf{n})\mathbf{e} = \mathbb{E}[G_k],$$

and if (56) satisfies for some $n_\Gamma(k)$,

$$(\boldsymbol{\alpha}_k \otimes \boldsymbol{\pi}) \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}| \leq n_\Gamma(k)}} \check{\Gamma}_k(\mathbf{n})\mathbf{e} > \mathbb{E}[G_k] - \frac{1}{2} \mathbb{E}[G_k(G_k - 1)] \varepsilon.$$

Lemma V.3 Suppose $\check{\mathbf{A}}_k(\mathbf{n})$ satisfies (34). Then there exists $n_\Gamma(k)$ satisfying (56).

Proof. From (54) and (55), it can be seen that $\check{\mathbf{A}}_k(\mathbf{n})$ ($|\mathbf{n}| \leq n_A(k)$) and $\check{\mathbf{\Gamma}}_k(\mathbf{n})$ ($\mathbf{n} \in \mathcal{Z}$) are related by

$$\sum_{\mathbf{n} \in \mathcal{Z}} \check{\mathbf{\Gamma}}_k(\mathbf{n}) = \sum_{m=0}^{\infty} \left(\mathbf{P}_k \otimes \sum_{\substack{\mathbf{l} \in \mathcal{Z} \\ |\mathbf{l}| \leq n_A(k)}} \check{\mathbf{A}}_k(\mathbf{l}) \right)^m.$$

Post-multiplying both sides of the above equation by $\mathbf{e} = \mathbf{e}(M_k) \otimes \mathbf{e}(M)$, we have

$$\begin{aligned} \sum_{\mathbf{n} \in \mathcal{Z}} \check{\mathbf{\Gamma}}_k(\mathbf{n}) \mathbf{e} &= \sum_{m=0}^{\infty} \left(\mathbf{P}_k \otimes \sum_{\substack{\mathbf{l} \in \mathcal{Z} \\ |\mathbf{l}| \leq n_A(k)}} \check{\mathbf{A}}_k(\mathbf{l}) \right)^m \cdot [\mathbf{e}(M_k) \otimes \mathbf{e}(M)] \\ &= \sum_{m=0}^{\infty} [\mathbf{P}_k^m \mathbf{e}(M_k)] \otimes \left[\left(\sum_{\substack{\mathbf{l} \in \mathcal{Z} \\ |\mathbf{l}| \leq n_A(k)}} \check{\mathbf{A}}_k(\mathbf{l}) \right)^m \mathbf{e}(M) \right]. \end{aligned}$$

Further, using (34), we obtain

$$\begin{aligned} \sum_{\mathbf{n} \in \mathcal{Z}} \check{\mathbf{\Gamma}}_k(\mathbf{n}) \mathbf{e} &> \sum_{m=0}^{\infty} (1 - \varepsilon)^m [\mathbf{P}_k^m \mathbf{e}(M_k)] \otimes \mathbf{e}(M) \\ &> \sum_{m=0}^{\infty} (1 - m\varepsilon) [\mathbf{P}_k^m \mathbf{e}(M_k)] \otimes \mathbf{e}(M) \\ &= \{(\mathbf{I} - \mathbf{P}_k)^{-1} \mathbf{e}(M_k)\} \otimes \mathbf{e}(M) - \varepsilon \{(\mathbf{I} - \mathbf{P}_k)^{-2} \mathbf{P}_k \mathbf{e}(M_k)\} \otimes \mathbf{e}(M), \end{aligned}$$

which completes the proof. \square

VI Numerical Examples

In this section, we show some numerical examples for queues with two arrival streams. Even though the algorithmic analysis has already been done for the single arrival cases [16, 17], no numerical examples were shown there. Thus the numerical result provided below is the first report in the literature, as for FIFO queues with Markovian arrival streams having different service time distributions.

In all numerical examples, the counting process of class k ($k = 1, 2$) arrivals follows a batch interrupted Poisson process with geometrically distributed batch size with mean g . Namely, the

counting process of class k ($k = 1, 2$) is characterized by $(\tilde{\mathbf{C}}_k, \tilde{\mathbf{D}}_k(n))$, where

$$\tilde{\mathbf{C}}_k = \begin{bmatrix} -2\lambda_k g^{-1} - 0.1 & 0.1 \\ 0.1 & -0.1 \end{bmatrix},$$

$$\tilde{\mathbf{D}}_k(n) = (1-p)p^{n-1} \begin{bmatrix} 2\lambda_k g^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad n = 1, 2, \dots,$$

where $p = 1 - 1/g$. Note that the arrival rate of class k is fixed to be λ_k regardless of the mean batch size g .

We now consider three types of the superposition of these two streams.

[Case P]

$$\mathbf{C} = \begin{bmatrix} -2(\lambda_1 + \lambda_2)g^{-1} - 0.1 & 0.1 \\ 0.1 & -0.1 \end{bmatrix},$$

and for $n = 1, 2, \dots$,

$$\mathbf{D}_1(n) = (1-p)p^{n-1} \begin{bmatrix} 2\lambda_1 g^{-1} & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{D}_2(n) = (1-p)p^{n-1} \begin{bmatrix} 2\lambda_2 g^{-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

[Case I]

$$\mathbf{C} = \tilde{\mathbf{C}}_1 \oplus \tilde{\mathbf{C}}_2,$$

and for $n = 1, 2, \dots$,

$$\mathbf{D}_1(n) = \tilde{\mathbf{D}}_1(n) \otimes \mathbf{I}(2), \quad \mathbf{D}_2(n) = \mathbf{I}(2) \otimes \tilde{\mathbf{D}}_2(n),$$

where \oplus denotes the Kronecker sum, and

[Case N]

$$\mathbf{C} = \begin{bmatrix} -2\lambda_1 g^{-1} - 0.1 & 0.1 \\ 0.1 & -2\lambda_2 g^{-1} - 0.1 \end{bmatrix},$$

and for $n = 1, 2, \dots$,

$$\mathbf{D}_1(n) = (1-p)p^{n-1} \begin{bmatrix} 2\lambda_1 g^{-1} & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{D}_2(n) = (1-p)p^{n-1} \begin{bmatrix} 0 & 0 \\ 0 & 2\lambda_2 g^{-1} \end{bmatrix}.$$

Note that in Case P, two arrival streams are positively correlated, in Case I, they are independent each other and in Case N, they are negatively correlated. As for the service time distributions,

we consider two cases, Case GD (class-dependent service times) and Case GI (i.i.d. service times):

[Case GD]

$$H_1 = 1, \text{ with prob. } 1, \quad H_2 = 4, \text{ with prob. } 1,$$

[Case GI]

$$H_k = \begin{cases} 1, & \text{with prob. } \lambda_1/(\lambda_1 + \lambda_2), \\ 4, & \text{with prob. } \lambda_2/(\lambda_1 + \lambda_2), \end{cases} \quad k = 1, 2,$$

where H_k ($k = 1, 2$) denotes a generic random variable for a service time of a class k customer. Note that the overall service time distributions are identical in both cases. We denote the queueing model with Case i ($i = P, I, N$) arrivals and Case j ($j = GD, GI$) services by Case (i, j) .

In what follows, we consider two examples, Examples 1 and 2, within the above settings. In Example 1, we set $\lambda_1 = \lambda_2 = 0.15$, so that $\rho_1 = 0.15$ and $\rho_2 = 0.6$ in Case (i, GD) ($i = P, I, N$), and $\rho_1 = \rho_2 = 0.375$ in Case (i, GI) ($i = P, I, N$). On the other hand, in Example 2, we set $\lambda_1 = 0.4$ and $\lambda_2 = 0.1$, so that $\rho_1 = \rho_2 = 0.4$ in Case (i, GD) ($i = P, I, N$) and that $\rho_1 = 0.64$ and $\rho_2 = 0.16$ in Case (i, GI) ($i = P, I, N$).

VI.1 Efficiency of the algorithm

Before showing the quantitative behavior of the queue length distribution, we discuss the efficiency of our numerical algorithm for the $\mathbf{F}_m(\mathbf{n})$. It follows from (19) that for $m = 0, 1, \dots$,

$$\sum_{\mathbf{n} \in \mathcal{Z}} \mathbf{F}_m(\mathbf{n}) = \left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \sum_{n_k=1}^{\infty} \mathbf{D}_k(n_k) \right) \right]^m, \quad (57)$$

where $\mathbf{I} + \theta^{-1}[\mathbf{C} + \sum_{k \in \mathcal{K}} \sum_{n_k=1}^{\infty} \mathbf{D}_k(n_k)]$ is a stochastic matrix. Thus a straightforward implementation of the recursion for the $\mathbf{F}_m(\mathbf{n})$ in (22) and (23) would be the following. We first truncate the $\mathbf{D}_k(n_k)$ at $n_k = n'_g(k)$ in such a way that

$$\theta^{-1} \sum_{n_k=1}^{n'_g(k)} \mathbf{D}_k(n_k) \mathbf{e} > \theta^{-1} \mathbf{D}_k \mathbf{e} - \frac{\varepsilon'_g}{K} \mathbf{e},$$

so that

$$\left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \sum_{n_k=1}^{n'_g(k)} \mathbf{D}_k(n_k) \right) \right] \mathbf{e} > (1 - \varepsilon'_g) \mathbf{e}.$$

We then compute all terms obtained by expanding the right hand side of (57) with the truncated $\mathbf{D}_k(n_k)$ ($k \in \mathcal{K}$). Note that if $\varepsilon'_g = \varepsilon_F$, the resulting $\mathbf{F}_m(\mathbf{n})$ satisfies (49) in Lemma V.2, where the summation on the left hand side of (49) is taken for all computed $\mathbf{F}_m(\mathbf{n})$'s.

Table 1: Number of computed $\mathbf{F}_m(\mathbf{n})$'s in Example 1.

		A : Our algorithm		B : Straightforward	
Case		$g = 1$	$g = 2$	$g = 5$	$g = 10$
(P, GD)	A	1.021×10^6	3.918×10^6	2.504×10^7	1.257×10^8
	B	2.453×10^6	2.107×10^9	3.696×10^{10}	4.243×10^{11}
(P, GI)	A	1.021×10^6	2.898×10^6	1.476×10^7	6.732×10^7
	B	2.453×10^6	1.438×10^9	1.812×10^{10}	1.825×10^{11}
(I, GD)	A	6.108×10^5	3.314×10^6	3.012×10^7	1.854×10^8
	B	1.993×10^6	2.919×10^9	9.286×10^{10}	1.649×10^{12}
(I, GI)	A	4.123×10^5	1.859×10^6	1.532×10^7	9.032×10^7
	B	1.253×10^6	1.292×10^9	3.741×10^{10}	6.173×10^{11}
(N, GD)	A	6.657×10^4	8.895×10^5	1.165×10^7	8.095×10^7
	B	1.378×10^5	2.620×10^8	1.066×10^{10}	1.955×10^{11}
(N, GI)	A	1.411×10^4	3.113×10^5	4.813×10^6	3.540×10^7
	B	2.743×10^4	7.158×10^7	3.278×10^9	6.442×10^{10}

In Table 1, we show the numbers of $\mathbf{F}_m(\mathbf{n})$'s computed by our algorithm and the above straightforward implementation, using Example 1, where we set $\varepsilon = 10^{-6}$, $\varepsilon_F = \text{r.h.s. of (36)} \times \varepsilon/2$ and $\varepsilon_g = \varepsilon_F/10$. We observe that for unbounded batch size cases (i.e., $g > 1$), the number of the computed $\mathbf{F}_m(\mathbf{n})$'s in our algorithm is less than that in the straightforward algorithm about by three order of magnitude. Thus, compared to the straightforward implementation, our algorithm is very efficient in terms of the computational time when the batch size is unbounded.

We note that a very huge memory space is required to store all $\check{\mathbf{F}}_m(\mathbf{n})$'s in some cases, even using our truncation and stopping criteria. For example, in Case (I, GD) with $g = 10$, the memory space to store all $\check{\mathbf{F}}_m(\mathbf{n})$'s is given by $16 \times 1.854 \times 10^8 \times 8 \text{ bytes} \approx 23.73 \text{ Gbytes}$, because each $\check{\mathbf{F}}_m(\mathbf{n})$ is a 4×4 matrix and one element requires 8 bytes in double precision. Thus in our implementation, every time $\check{\mathbf{F}}_m(\mathbf{n})$'s for each m are obtained, we compute the contributions of $\check{\mathbf{F}}_m(\mathbf{n})$'s to $\check{\mathbf{A}}_k(\mathbf{n})$ and $\check{\mathbf{v}}_k(\mathbf{n})$ in Step (3-c), and discard all $\check{\mathbf{F}}_{m-1}(\mathbf{n})$'s.

Table 2 shows the maximum number of $\check{\mathbf{F}}_m(\mathbf{n})$'s stored temporarily in our algorithm, where the ratio of it to the total number of computed $\check{\mathbf{F}}_m(\mathbf{n})$'s is also shown in parenthesis. We observe that in most cases, the number of temporarily stored $\check{\mathbf{F}}_m(\mathbf{n})$'s is a few percent of the total number of computed ones. Thus our implementation is expected to save the required memory space, especially when a large number of $\check{\mathbf{F}}_m(\mathbf{n})$'s should be computed.

Table 2: Number of stored $\check{F}_m(\mathbf{n})$'s in Example 1.

Case	$g = 1$	$g = 2$	$g = 5$	$g = 10$
(P, GD)	27225 (2.67%)	90601 (2.31%)	455625 (1.82%)	1651227 (1.31%)
(P, GI)	27225 (2.67%)	75076 (2.59%)	330051 (2.24%)	1125723 (1.67%)
(I, GD)	16641 (2.72%)	68121 (2.06%)	399424 (1.33%)	1548781 (0.84%)
(I, GI)	13110 (3.18%)	47524 (2.56%)	263683 (1.72%)	997003 (1.10%)
(N, GD)	4970 (7.47%)	41209 (4.63%)	324331 (2.78%)	1387686 (1.71%)
(N, GI)	1764 (12.51%)	21171 (6.80%)	187491 (3.90%)	833571 (2.35%)

VI.2 Number of customers in Example 1

Figures 1–3 plot the complementary distributions of the total number N of customers in Case (i, GD) and Case (i, GI) ($i = \text{P, I, N}$), where the batch size is fixed to be one, i.e., $g = 1$. Note that the overall input processes in Case (P, GD) and Case (P, GI) are identical, so that the distributions of the total number of customers are also identical, as shown in Figure 1. However, as shown in Table 3, the joint queue length distributions in these two cases are different. Note also that in Case (P, GI), $\mathbf{p}(n_1, n_2)\mathbf{e} = \mathbf{p}(n_2, n_1)\mathbf{e}$, because the conditional joint distribution $\Pr(N_1 = n_1, N_2 = n_2 \mid N_1 + N_2 = n_1 + n_2)$ follows a binomial distribution with parameter 0.5. We also observe that $\mathbf{p}(n, n)\mathbf{e}$'s in both cases take the same value for each n . Unfortunately, we cannot provide any intuitive explanation of this phenomenon.

From Figures 2 and 3, we observe that class-dependent service times cause longer tails in the total queue length distributions, in these specific examples. We shall explain this phenomenon for Case N. In Case (N, GD), the conditional expected amounts of work brought into the system per unit time given the state of the underlying Markov chain are different, and they are given by 0.3 and 1.2, respectively. Thus in Case (N, GD), the system is overloaded during a half of time. On the other hand, in Case (N, GI), the conditional expected amount of work brought into the system per unit time is fixed to be 0.75, regardless of the state of the underlying Markov chain. Therefore the distribution of the total number of customers in Case (N, GD) has a longer tail than that in Case (N, GI).

Next, we consider the expected total number $E[N]$ of customers as a function of the mean

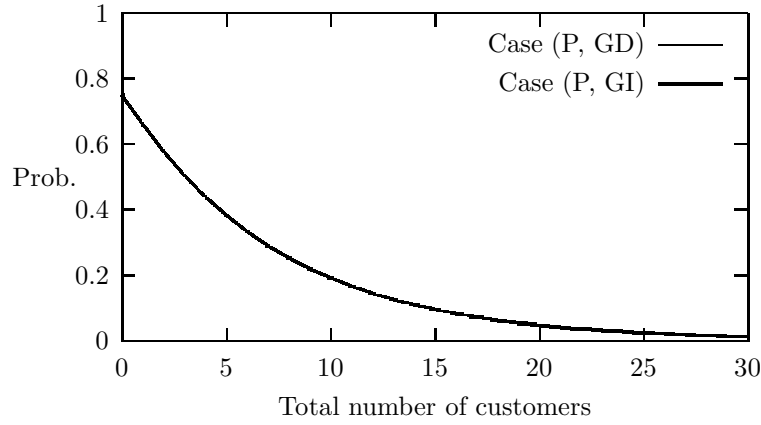


Figure 1: Complementary distribution of total number of customers in Example 1.

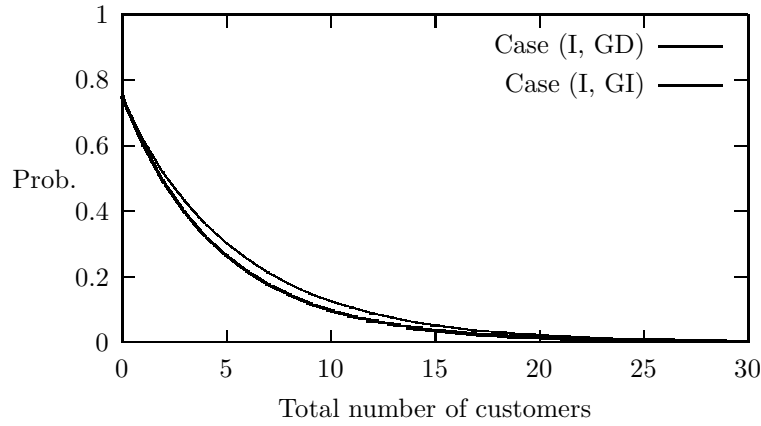


Figure 2: Complementary distribution of total number of customers in Example 1.

batch size g . Table 4 shows $E[N]$ for the mean batch size $g = 1, 2, 3, 4, 5$ and 10 . We observe that $E[N]$ increases with the mean batch size g in all cases. This phenomenon comes from the fact that the deviation of the amount of work brought into the system per unit time increases with g . We also observe that for a fixed g , the positive correlation in the two streams leads to a larger $E[N]$ in both Cases GD and GI, as expected.

VI.3 Number of customers in Example 2

Table 5 shows the expected total number $E[N]$ of customers for the mean batch size $g = 1, 2, 3, 4, 5$ and 10 . We first examine the case of $g = 1$. Contrary to Example 1, we observe that the class-dependent service time (Case GD) decreases the expected total number of customers in Cases I and N. This phenomenon can be explained in a similar way to Example 1. For example,

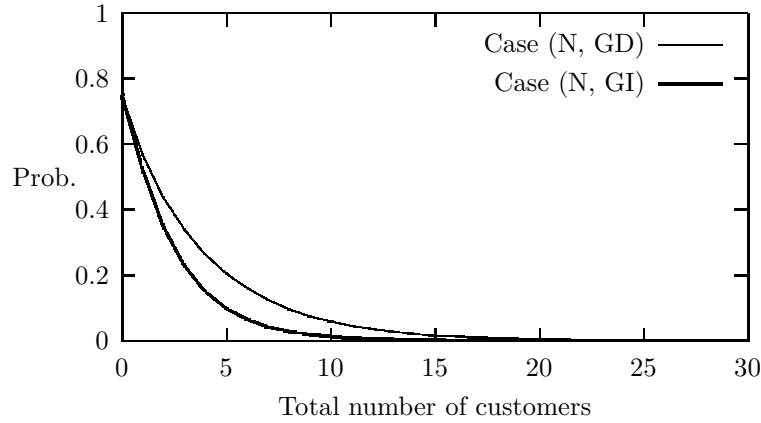


Figure 3: Complementary distribution of total number of customers in Example 1.

in Case (N, GD), the conditional expected amount of work brought into the system per unit time is fixed to be 0.8, regardless of the state of the underlying Markov chain. On the other hand, in Case (N, GI), the conditional expected amounts of work brought into the system per unit time given the state of the underlying Markov chain are different, and they are given by 1.28 and 0.32, respectively. Thus in Case (N, GI), the system is overloaded during a half of time, so that $E[N]$ in Case (N, GI) is greater than that in Case (N, GD).

We observe that in any case, the expected total number of customer increases with the mean batch size g , as in Example 1, and that $E[N]$ in Case GD eventually becomes greater than $E[N]$ in Case GI. We also observe that for a fixed g , the positive correlation in the two streams leads to a larger $E[N]$ in both Cases GD and GI, as in Example 1.

VII Concluding Remarks

We developed a numerically feasible procedure to compute the joint queue length distribution in a FIFO single-server queue with multiple batch Markovian arrival streams, under the assumption that service time distributions of customers from respective arrival streams are different and the batch size distributions follow discrete phase-type distributions. We established several truncation and stopping criteria to ensure the numerical accuracy in the final result.

Note, however, that the computation of the joint queue length distribution is intensive by nature, especially when the number of classes is large. Even in such a case, the steady state distribution of the total number of customers can be readily computed by modifying our algorithm. For the sake of completeness, we show algorithm steps for the total number of customers in Appendix. Note here that the algorithm to compute $A_k^{(T)}(n)$ in (58) for the number of arrivals in a service time can be used in the standard algorithm for the ordinary BMAP/GI/1 queue, too

Table 3: Joint queue length distribution $\mathbf{p}(n_1, n_2)e$.
(Upper rows for Case (P, GD) and lower rows for Case (P, GI))

n_1	0	1	2	3
n_2				
0	2.500×10^{-1}	2.472×10^{-2}	8.593×10^{-3}	3.481×10^{-3}
	2.500×10^{-1}	4.501×10^{-2}	2.054×10^{-2}	9.224×10^{-3}
1	6.530×10^{-2}	4.108×10^{-2}	2.193×10^{-2}	1.118×10^{-2}
	4.501×10^{-2}	4.108×10^{-2}	2.767×10^{-2}	1.629×10^{-2}
2	3.249×10^{-2}	3.341×10^{-2}	2.444×10^{-2}	
	2.054×10^{-2}	2.767×10^{-2}	2.444×10^{-2}	
3	1.497×10^{-2}	2.141×10^{-2}		
	9.224×10^{-3}	1.629×10^{-2}		
4	6.630×10^{-3}			
	4.073×10^{-3}			

Table 4: Expected total number of customers in Example 1.

Case	$g = 1$	$g = 2$	$g = 3$	$g = 4$	$g = 5$	$g = 10$
(P, GD)	5.8760	9.9815	13.9356	17.8320	21.7001	40.8865
(P, GI)	5.8760	9.1466	12.2898	15.3793	18.4408	33.5873
(I, GD)	4.5417	8.5777	12.4865	16.3524	20.1987	39.3295
(I, GI)	4.0010	7.1857	10.2714	13.3219	16.3555	31.4326
(N, GD)	3.2822	7.2033	11.0527	14.8822	18.7035	37.7739
(N, GI)	2.2800	5.2800	8.2800	11.2800	14.2800	29.2800

(see [7, 14]), because the BMAP/GI/1 queue is considered as a special case of $K = 1$ and the sequence of matrices for the number of arrivals in a service time is essential for computing the queue length distribution. To the best of our knowledge, however, there is no work to consider the truncation and stopping criterion to compute $\mathbf{A}_k^{(T)}(n)$ in the BMAP/GI/1 queue. Thus our development also contributes to the standard algorithm for the BMAP/GI/1 queue.

Appendix: Algorithm for the Total Number of Customers

We show a numerical algorithm to compute the steady state distribution of the total number of customers, by modifying our algorithm for the joint queue length distribution. We define

Table 5: Expected total number of customers in Example 2.

Case	$g = 1$	$g = 2$	$g = 3$	$g = 4$	$g = 5$	$g = 10$
(P, GD)	11.5019	17.7712	23.8347	29.8053	35.7261	65.0310
(P, GI)	11.5019	15.9366	20.1933	24.3657	28.4904	48.8117
(I, GD)	7.1517	13.3270	19.3007	25.2050	31.0760	60.2474
(I, GI)	8.7304	13.1052	17.3093	21.4407	25.5333	45.7640
(N, GD)	3.2168	9.0326	14.8425	20.6497	26.4551	55.4705
(N, GI)	6.0892	10.3399	14.4641	18.5407	22.5933	42.7206

$\mathbf{p}^{(T)}(n)$ ($n = 0, 1, \dots$) and $\mathbf{q}_k^{(T)}(n)$ ($k \in \mathcal{K}, n = 0, 1, \dots$) as

$$\mathbf{p}^{(T)}(n) = \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}|=n}} \mathbf{p}(\mathbf{n}), \quad \mathbf{q}_k^{(T)}(n) = \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}|=n}} \mathbf{q}_k(\mathbf{n}),$$

respectively. Corollary IV.1 is then reduced to

$$\begin{aligned} \mathbf{p}^{(T)}(0) &= \sum_{k \in \mathcal{K}} \lambda_k \mathbf{q}_k^{(T)}(0) (-\mathbf{C})^{-1}, \\ \mathbf{p}^{(T)}(n) &= \sum_{k \in \mathcal{K}} \left[\lambda_k \left(\mathbf{q}_k^{(T)}(n) - \mathbf{q}_k^{(T)}(n-1) \right) \right. \\ &\quad \left. + \sum_{m=1}^n \mathbf{p}^{(T)}(n-m) \mathbf{D}_k(m) \right] (-\mathbf{C})^{-1}, \quad n = 1, 2, \dots \end{aligned}$$

Further, under Assumption IV.1, Theorem IV.3 is reduced to

$$\begin{aligned} \mathbf{q}_k^{(T)}(n) &= \frac{1}{\lambda_k} \sum_{\substack{m_1+m_2+m_3 \\ +m_4=n}} \mathbf{v}_k^{(T)}(m_1) [\boldsymbol{\alpha}_k \otimes \mathbf{A}_k^{(T)}(m_2)] \\ &\quad \cdot \boldsymbol{\Gamma}_k^{(T)}(m_3) [\{\mathbf{P}_k^{m_4}(\mathbf{I} - \mathbf{P}_k)\mathbf{e}\} \otimes \mathbf{I}(M)], \end{aligned}$$

where

$$\begin{aligned} \mathbf{A}_k^{(T)}(n) &= \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}|=n}} \mathbf{A}_k(\mathbf{n}), \\ \mathbf{v}_k^{(T)}(n) &= \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}|=n}} \mathbf{v}_k(\mathbf{n}), \quad \boldsymbol{\Gamma}_k^{(T)}(n) = \sum_{\substack{\mathbf{n} \in \mathcal{Z} \\ |\mathbf{n}|=n}} \boldsymbol{\Gamma}_k(\mathbf{n}). \end{aligned} \tag{58}$$

Thus the $\mathbf{p}^{(T)}(n)$ is obtained if we compute the $\mathbf{A}_k^{(T)}(n)$, the $\mathbf{v}_k^{(T)}(n)$ and the $\boldsymbol{\Gamma}_k^{(T)}(n)$.

Note here that $\mathbf{A}_k^{(\text{T})}(n)$, $\mathbf{v}_k^{(\text{T})}(n)$ and $\mathbf{\Gamma}_k^{(\text{T})}(n)$ satisfy

$$\begin{aligned}\sum_{n=0}^{\infty} z^n \mathbf{A}_k^{(\text{T})}(n) &= \int_0^{\infty} dH_k(x) \exp \left[\left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z) \right) x \right], \\ \sum_{n=0}^{\infty} z^n \mathbf{v}_k^{(\text{T})}(n) &= \int_0^{\infty} d\mathbf{v}(x) \mathbf{D}_k \exp \left[\left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z) \right) x \right], \\ \sum_{n=0}^{\infty} z^n \mathbf{\Gamma}_k^{(\text{T})}(n) &= \left[\mathbf{I} - \mathbf{P}_k \otimes \int_0^{\infty} dH_k(x) \exp \left[\left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z) \right) x \right] \right]^{-1},\end{aligned}$$

respectively. Thus $\mathbf{A}_k^{(\text{T})}(n)$ and $\mathbf{v}_k^{(\text{T})}(n)$ can be written to be

$$\begin{aligned}\mathbf{A}_k^{(\text{T})}(n) &= \sum_{m=0}^{\infty} \gamma_k^{(m)}(\theta) \mathbf{F}_m^{(\text{T})}(n), \\ \mathbf{v}_k^{(\text{T})}(n) &= \sum_{m=0}^{\infty} \mathbf{v}^{(m)}(\theta) \mathbf{D}_k \mathbf{F}_m^{(\text{T})}(n),\end{aligned}$$

respectively, where $\mathbf{F}_m^{(\text{T})}(n)$ denotes an $M \times M$ matrix which satisfies

$$\sum_{n=0}^{\infty} z^n \mathbf{F}_m^{(\text{T})}(n) = \left[\mathbf{I} + \theta^{-1} \left(\mathbf{C} + \sum_{k \in \mathcal{K}} \mathbf{D}_k^*(z) \right) \right]^m.$$

Further the $\mathbf{\Gamma}_k^{(\text{T})}(n)$ ($k \in \mathcal{K}$, $n \geq 0$) is determined by the following recursion:

$$\mathbf{\Gamma}_k^{(\text{T})}(0) = \left[\mathbf{I} - \mathbf{P}_k \otimes \mathbf{A}_k^{(\text{T})}(0) \right]^{-1},$$

and for $n = 1, 2, \dots$,

$$\mathbf{\Gamma}_k^{(\text{T})}(n) = \sum_{l=1}^n \mathbf{\Gamma}_k^{(\text{T})}(n-l) \left[\mathbf{P}_k \otimes \mathbf{A}_k^{(\text{T})}(l) \right] \mathbf{\Gamma}_k^{(\text{T})}(0).$$

Thus we can compute $\mathbf{A}_k^{(\text{T})}(n)$, $\mathbf{v}_k^{(\text{T})}(n)$ and $\mathbf{\Gamma}_k^{(\text{T})}(n)$ by replacing Steps 3 and 4 with the followings.

Step 3. Compute $\check{\mathbf{A}}_k^{(\text{T})}(n)$ and $\check{\mathbf{v}}_k^{(\text{T})}(n)$ by the following procedure, where the initial values of $\check{\mathbf{A}}_k^{(\text{T})}(n)$ and $\check{\mathbf{v}}_k^{(\text{T})}(n)$ ($n \geq 0$) are assumed to be \mathbf{O} and $\mathbf{0}$, respectively.

Step (3-a). Set $\check{\mathbf{F}}_0^{(\text{T})}(0) = \mathbf{I}$ and $n_F^{(0)} = 0$. Also set

$$\check{\mathbf{A}}_k^{(\text{T})}(0) = \gamma_k^{(0)}(\theta) \mathbf{I}, \quad \check{\mathbf{v}}_k^{(\text{T})}(0) = \mathbf{v}^{(0)}(\theta) \mathbf{D}_k, \quad \forall k \in \mathcal{K}.$$

Step (3-b). Set $n_F^{(1)} = \max_{k \in \mathcal{K}} n_g(k)$ and $m = 1$, and compute $\check{\mathbf{F}}_1^{(T)}(n)$ by the following recursion:

$$\check{\mathbf{F}}_1^{(T)}(0) = \mathbf{I} + \theta^{-1}\mathbf{C},$$

and for $n = 1, 2, \dots, n_F^{(1)}$,

$$\check{\mathbf{F}}_1^{(T)}(n) = \theta^{-1} \sum_{k \in \mathcal{K}} U(n_g(k) - n) g_k(n) \mathbf{D}_k.$$

Step (3-c). For each $k \in \mathcal{K}$, if $m \leq m_\gamma(k)$, add $\gamma_k^{(m)}(\theta) \check{\mathbf{F}}_m^{(T)}(n)$ to $\check{\mathbf{A}}_k^{(T)}(n)$ for all $n \leq n_F^{(m)}$. Also, for each $k \in \mathcal{K}$, if $m \leq m_v(k)$, add $\mathbf{v}^{(m)}(\theta) \mathbf{D}_k \check{\mathbf{F}}_m^{(T)}(n)$ to $\check{\mathbf{v}}_k^{(T)}(n)$ for all $n \leq n_F^{(m)}$.

Step (3-d). If $m \geq m_{\max}$, stop computing, and otherwise, add one to m and go to Step (3-e).

Step (3-e). For each $n = 0, 1, \dots$, compute $\check{\mathbf{F}}_m^{(T)}(n)$ by

$$\begin{aligned} \check{\mathbf{F}}_m^{(T)}(n) &= U\left(n_F^{(m-1)} - n\right) \check{\mathbf{F}}_{m-1}^{(T)}(n) (\mathbf{I} + \theta^{-1}\mathbf{C}) \\ &\quad + \sum_{l=1}^{\min(n, n_F^{(1)})} U\left(n_F^{(m-1)} - n + l\right) \check{\mathbf{F}}_{m-1}^{(T)}(n-l) \check{\mathbf{F}}_1^{(T)}(l), \end{aligned}$$

until $\check{\mathbf{F}}_m^{(T)}(n)$'s satisfy $\sum_{n \leq n^*} \check{\mathbf{F}}_m^{(T)}(n) \mathbf{e} > (1 - \varepsilon_F)^m \mathbf{e}$ for some n^* . Let $n_F^{(m)} = n^*$ and go to Step (3-c).

Step 4. Set

$$n_A(k) = \max\left(n_F^{(m)}; m = 0, 1, \dots, m_\gamma(k)\right),$$

and for each $k \in \mathcal{K}$, compute $\check{\mathbf{\Gamma}}_k^{(T)}(n)$ by the following recursion:

$$\check{\mathbf{\Gamma}}_k^{(T)}(0) = \left[\mathbf{I} - \mathbf{P}_k \otimes \check{\mathbf{A}}_k^{(T)}(0)\right]^{-1},$$

and for $n = 1, 2, \dots$,

$$\check{\mathbf{\Gamma}}_k^{(T)}(n) = \sum_{l=1}^n U(n_A(k) - l) \check{\mathbf{\Gamma}}_k^{(T)}(n-l) \left[\mathbf{P}_k \otimes \check{\mathbf{A}}_k^{(T)}(l)\right] \check{\mathbf{\Gamma}}_k^{(T)}(0),$$

until $\check{\mathbf{\Gamma}}_k^{(T)}(n)$'s satisfy

$$\sum_{n=0}^{n_\Gamma(k)} \check{\mathbf{\Gamma}}_k^{(T)}(n) \mathbf{e} > \{(\mathbf{I} - \mathbf{P}_k)^{-1} \mathbf{e}(M_k)\} \otimes \mathbf{e}(M) - \varepsilon \{(\mathbf{I} - \mathbf{P}_k)^{-2} \mathbf{P}_k \mathbf{e}(M_k)\} \otimes \mathbf{e}(M),$$

for some integer $n_\Gamma(k)$.

Remark A.1 The above algorithm ensures that

$$\sum_{n=0}^{n_A(k)} \check{\mathbf{A}}_k^{(T)}(n) \mathbf{e} > (1 - \varepsilon) \mathbf{e}, \quad \sum_{n=0}^{n_v(k)} \check{\mathbf{v}}_k^{(T)}(n) \mathbf{e} > (1 - \varepsilon) \lambda_k^{(B)},$$

respectively, where $n_v(k)$ is given by

$$n_v(k) = \max \left(n_F^{(m)}; m = 0, 1, \dots, m_v(k) \right).$$

Further $\check{\mathbf{\Gamma}}_k^{(T)}(n)$ satisfies

$$(\alpha_k \otimes \pi) \sum_{n=0}^{n_\Gamma(k)} \check{\mathbf{\Gamma}}_k^{(T)}(n) \mathbf{e} > \mathbb{E}[G_k] - \frac{1}{2} \mathbb{E}[G_k(G_k - 1)] \varepsilon.$$

References

- [1] Asmussen, S. Ladder Heights and the Markov-Modulated $M/G/1$ Queue. *Stoch. Proc. Appl.* **1991**, 37, 313–326.
- [2] Asmussen, S.; Koole, G. Marked Point Processes as Limits of Markovian Arrival Streams. *J. Appl. Prob.* **1993**, 30, 365–372.
- [3] He, Q.-M. Queues with Marked Customers. *Adv. Appl. Prob.* **1996**, 28, 567–587.
- [4] He, Q.-M. The Versatility of MMAP[K] and the MMAP[K]/G[K]/1 Queue. *QUESTA* **2001**, 38, 397–418.
- [5] Loynes, R. M. The Stability of a Queue with Non-Independent Interarrival and Service Times. *Proc. Cambridge Philos. Soc.* **1962**, 58, 497–520.
- [6] Lucantoni, D. M.; Meier-Hellstern, K. S.; Neuts, M. F. A Single-Server Queue with Server Vacations and a Class of Non-renewal Arrival Processes. *Adv. Appl. Prob.* **1990**, 22, 676–705.
- [7] Lucantoni, D. M. New Results on the Single Server Queue with a Batch Markovian Arrival Process. *Stoch. Mod.* **1991**, 7, 1–46.
- [8] Machihara, F. A BMAP/SM/1 Queue with Service Times Depending on the Arrival Process. *QUESTA* **1999**, 33, 277–291.
- [9] Neuts, M. F. 1989; *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekker: New York.
- [10] Regterschot, G. J. K.; de Smit, J. H. A. The Queue $M|G|1$ with Markov Modulated Arrivals and Services. *Math. Oper. Res.* **1986**, 11, 465–483.

- [11] Takine, T.; Hasegawa, T. The Workload in the MAP/G/1 Queue with State-dependent Services: Its Application to a Queue with Preemptive Resume Priority. *Stoch. Mod.* **1994**, *10*, 183–204.
- [12] Takine, T.; Matsumoto, Y.; Suda, T.; Hasegawa, T. Mean Waiting Times in Nonpreemptive Priority Queues with Markovian Arrival and i.i.d. Service Processes. *Perfor. Eval.* **1994**, *20*, 131–149.
- [13] Takine, T. A Continuous Version of Matrix-Analytic Methods with the Skip-Free to the Left Property. *Stoch. Mod.* **1996**, *12*, 673–682.
- [14] Takine, T. A New Recursion for the Queue Length Distribution in the Stationary BMAP/GI/1 Queue. *Stoch. Mod.* **2000**, *16*, 335–341.
- [15] Takine, T. A Recent Progress in Algorithmic Analysis of FIFO Queues with Markovian Arrival Streams. *J. Korean Math. Soc.* **2001**, *38*, 807–842.
- [16] Takine, T. Distributional Form of Little's Law for FIFO Queues with Multiple Markovian Arrival Streams and Its Application to Queues with Vacations. *QUESTA* **2001**, *37*, 31–63.
- [17] Takine, T. Queue Length Distribution in a FIFO Single-Server Queue with Multiple Arrival Streams Having Different Service Time Distributions. *QUESTA* **2001**, *39*, 349–375.
- [18] Zhu, Y; Prabhu, N. U. Markov-Modulated $PH/G/1$ Queueing Systems. *QUESTA* **1991**, *9*, 313–322.